OHM 电子爱好者读物

机电一体化入门

〔日〕岩本 洋 主编 森田克己 天野一美 著



科学出版社
www.sciencep.com

(TN-0400.0101)
责任编辑　崔炳哲
责任制作　魏　谨
责任印制　刘士平
封面设计　李　力

◇◇◇◇◇◇◇ 内　容　简　介 ◇◇◇◇◇◇◇

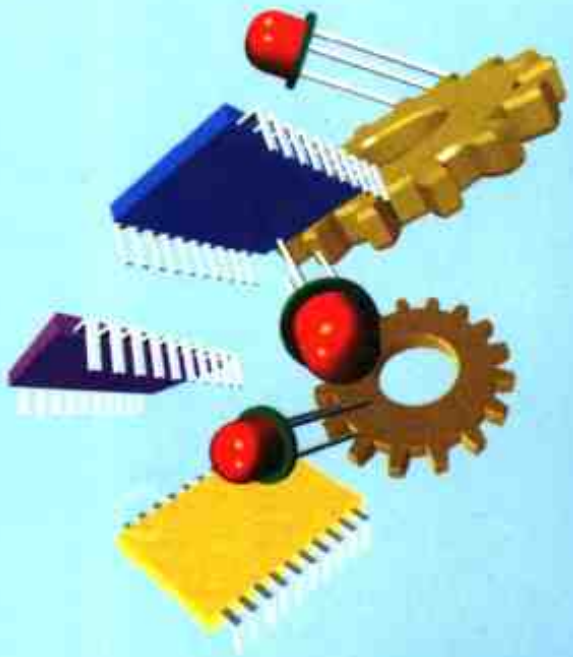
“OHM电子爱好者读物”系列共有6本，涉及电子电路、电子机械、电子控制、数字电路、机电一体化、电子技术等。本系列以初学者为对象，以实用技术为重点，利用丰富的插图，甚至建立模拟教室以对话的形式、通俗易懂地介绍相关内容。阅读本系列的读者最好边学习边实践，以使所学知识变成自己有用的技能。

本书是“OHM电子爱好者读物”之一，内容包括机电一体化的控制装置，传感器、执行机构、控制等基础知识，以及计算机与接口，输入输出装置与接口，简单的控制程序，简单装置的控制等；最后一章作为实践部分，专门介绍相扑机器人的制作技巧。

本书适合作为职业技术类教材，亦可作为企业技术人员的初级培训用教材。

OHM 电子爱好者读物

电子电路入门
数字电路入门
电子机械入门
电子电路与电子技术入门
电子控制入门
机电一体化入门



ISBN 7-03-010374-2
9 787030 103741 >

TH-39
2Y17

ISBN 7-03-010374-2
定　价: 24.00 元

OHM电子爱好者读物

机电一体化入门

〔日〕岩本　洋　主编　森田克己　天野一美　著
徐其荣　译
郑佩玉　校

A1003091

科学出版社
北京

图字：01-2002-1666 号

Original Japanese language edition
Hajimete Manabu Mechatronics Nyumon Hayawakari
By Hiroshi Iwamoto, Katsumi Morita and Kazumi Amano
Copyright © 1997 by Hiroshi Iwamoto, Katsumi Morita and Kazumi Amano
Published by Ohmsha, Ltd.
This Chinese version published by Science Press, Beijing
Under license from Ohmsha, Ltd.
Copyright © 2002
All rights reserved

初めて学ぶ
メカトロニクス入門早わかり
岩本　洋　森田克己　天野一美　オーム社　1997

图书在版编目(CIP)数据

机电一体化入门/(日)岩本 洋主编,森田克己,天野一美著;徐其荣译.
北京:科学出版社,2003
(OHM 电子爱好者读物)
ISBN 7-03-010374-2
Ⅰ.机… Ⅱ.①岩…②森…③天…④徐… Ⅲ.机电一体化 Ⅳ.TH-39
中国版本图书馆 CIP 数据核字(2002)第 026229 号

责任编辑　崔炳哲　　责任制作　魏　谨
责任印制　刘士平　　封面设计　李　力

科学出版社　出版
北京东黄城根北街 16 号　邮政编码:100717
http://www.sciencep.com
深海印刷有限责任公司 印刷
北京东方科龙图文有限公司　制作
http://www.okbook.com.cn
科学出版社发行　各地新华书店经销
2003 年 5 月第　一　版　开本:B5(720×1000)
2003 年 5 月第一次印刷　印张:15
印数:1—5 000　　字数:260 000
定　价:24.00 元
(如有印装质量问题,我社负责调换(新欣))

前言

在产业界，尤其是在制造业中的所有领域，均在追求自动化与信息化。其目的自然是为了节约劳动力、提高产品合格率与劳动生产率。

要推进自动化，机械学（mechanics）与电子学（electronics）二者缺一不可。把二者结合在一起，作为一个术语就称为“机电一体化”（mechatronics）。但“mechatronics”不是规范的英语，是日本造出来的，即所谓的日本式英语。

对“机电一体化”稍加思索就可知道，除了机械技术与电子技术之外，它还与信息技术密切相关，将这些技术综合为一体就构成了机电一体化。也就是说，机电一体化技术是由核心技术与其他多种技术组合而成的。其核心技术有：把物理量或化学量变换为电气量的传感器技术，能产生旋转运动或直线运动的执行机构技术，连接传感器与计算机或连接计算机与执行机构的接口技术等；其他多种技术则包括控制技术中的顺序控制技术与反馈控制技术和能操纵计算机的程序技术等。

本书作为技术入门书，考虑到初学机电一体化的读者，在撰稿时，我们注重了以下几点：

1. 内容表达尽量做到通俗易懂。

2. 尽量利用插图帮助读者直观地理解相关内容。

3. 以学习机电一体化的基础知识作为出发点，不详述学术理论方面的细节，而以实用为主。

4. 每二三页介绍一个主题，最后通过解习题加深对正文的理解。

本书的执笔分工如下：

第1～4章　东京都立藏前工业高等学校　森田克己

第5～8章　东京都立工业技术教育中心　天野一美

希望读者通过对本书的学习，能掌握机电一体化的基础知识，作为进一步深造的台阶。

全国工业高等学校校长协会　顾问　岩本　洋

前言

进入21世纪，尤其我国加入WTO之后，科学技术、工农业生产迎来了新的发展机遇，也面临着新的挑战。

为了适应在国际大市场上竞争的需要，各行各业的人们都迫切需要学习新技术、新理论、新方法。机电一体化正是所有人都多少会接触到的实用技术。它广泛深入到从军用到民用、从工业到农业的所有国民经济领域，小到手表、照相机，大到汽车、飞机、火箭、人造卫星、核电站等，无不应用了机电一体化的技术和产品。

《机电一体化》这本书，是满足各行各业人们需要的一本入门教材。书中对原理的讲述深入而详细，图文并茂，文字浅显易懂，并有练习题、挑战题，便于复习掌握要点。非常适合作为中高级技师、就业培训及军地两用人才的培训教材，也可以供有初中以上文化程度的有兴趣的读者自学之用。也是理工科中等专业学校的教师们了解这方面知识的必备教材或参考手册。本书的编写既全面地描述了机电一体化的各方面技术，充分阐述其基本原理，又有非常实用的例子，适合实际制作，并且各章节中一个一个专题均自成体系，相对独立。适合于不同层次的读者任意选学。由于本书的上述特色，其原版书十分畅销并已多次重印，深受读者欢迎。

本书中所提及的内容，既有当前正在普及的技术，也有一些比较新颖的内容，如最后部分介绍的机器人的制作等。读者可在掌握本书内容的基础上，进一步学习自己感兴趣的相关新技术，努力跟上时代的前进步伐。

目　录



第1章 机电一体化的控制装置 1

1 机电一体化的支持技术 …… 2
2 常见的机电一体化 …… 5
3 机电一体化的机械装置 …… 8
4 机电一体化的电路知识(1) …… 11
5 机电一体化的电路知识(2) …… 14
6 可编程序控制器 …… 17
7 PC的编程 …… 20
◆ 挑战题 …… 23

第2章 传感器的基础知识 24

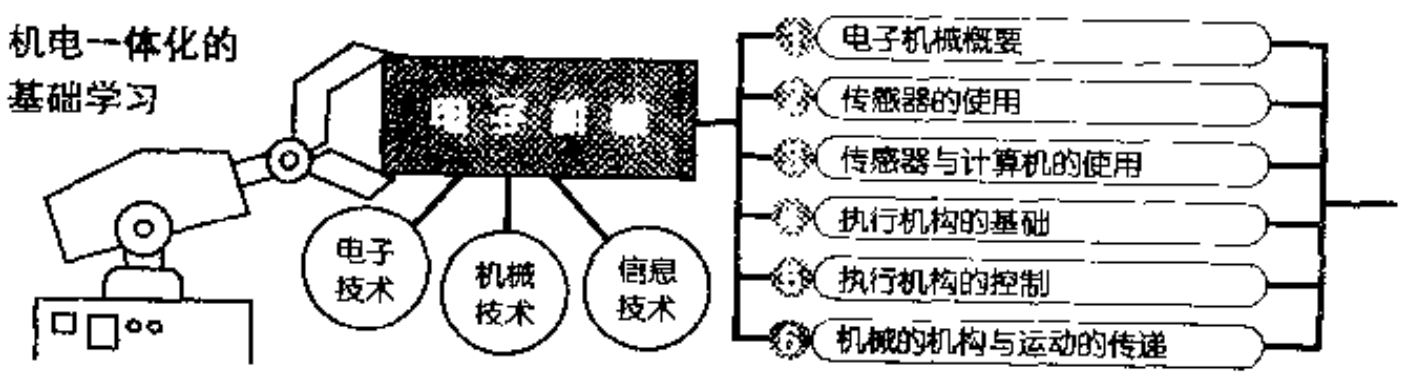
1 传感器的定义 …… 25
2 各种各样的传感器(1) …… 28
3 各种各样的传感器(2) …… 31
4 来自传感器的输入信号 …… 34
5 数字量的输入 …… 37
6 模拟量的输入 …… 40
7 传感器输入实践 …… 43
◆ 挑战题 …… 46

第3章 执行机构的基础知识 47

1 执行机构的作用 …… 48
2 执行机构的驱动电路 …… 51
3 继电器电路 …… 54
4 使用晶体管的驱动电路 …… 57
5 螺线管驱动电路 …… 60
6 步进电动机的驱动电路 …… 63
7 固体继电器的驱动电路 …… 66
◆ 挑战题 …… 69

第4章 控制的基础知识 70

1 控制的定义 …… 71
2 计算机控制的定义 …… 74
3 数字信号的定义 …… 77
4 逻辑电路实验装置的制作方法 …… 80
5 IC实验(1)　AND电路的实验 … 83
6 IC实验(2)　OR、NOT电路的实验 …… 86
7 IC实验(3)　NAND、NOR电路的实验 …… 89
◆ 挑战题 …… 92



第5章 计算机与接口 93

1 计算机的构造 …… 94
2 计算机的信号(1) …… 97
3 计算机的信号(2) …… 101
4 接口的定义 …… 104
5 数据传输标准与通用接口 …… 107
6 信号与程序 …… 110
7 用于控制的主要程序语言 …… 113
◆ 挑战题 …… 116

第6章 输入输出装置与接口 117

1 8255 输入输出接口管脚与使用模式 118
2 试制 8255 输入输出接口板(1) …… 121
3 试制 8255 输入输出接口板(2) …… 124
4 8 位发光二极管的点灯电路 …… 127
5 开关用接口电路 …… 130
6 电磁继电器与接口 …… 133
7 小型直流电动机的接口 …… 136
◆ 挑战题 …… 139

第7章 简单的控制程序 140

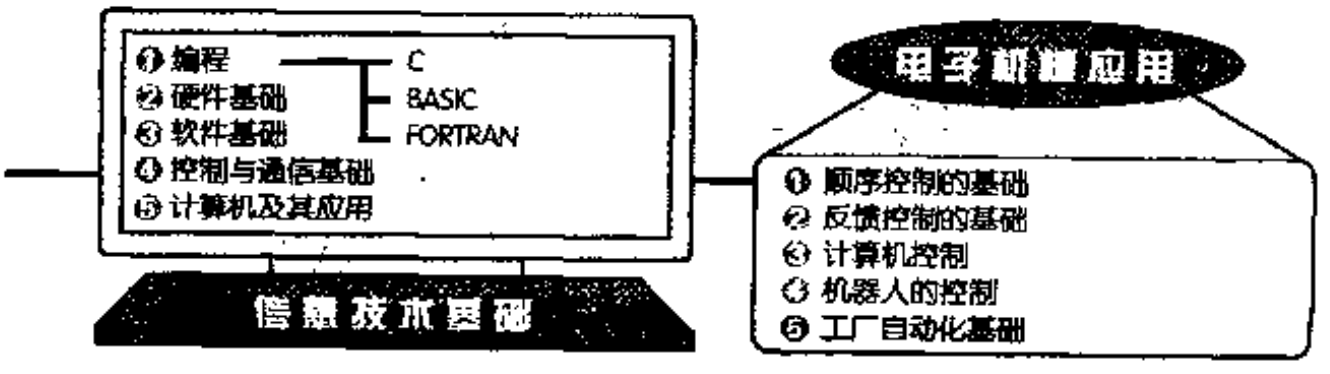
1 在控制中使用 C 语言的基本知识 … 141
2 8 位发光二极管的亮、熄程序 …… 145
3 利用开关的控制 …… 148
4 利用传感器的控制 …… 151
5 汽缸的控制 …… 154
6 直流电动机的控制 …… 157
7 步进电动机的控制 …… 160
◆ 挑战题 …… 163

第8章 简单装置的控制 164

1 用气动执行机构的传送装置的控制 …… 165
2 简单自动门的控制 …… 175
3 用步进电动机的传送装置的控制 … 183
4 铁路模型的控制 …… 195
5 中断控制的设计 …… 206
◆ 挑战题 …… 209

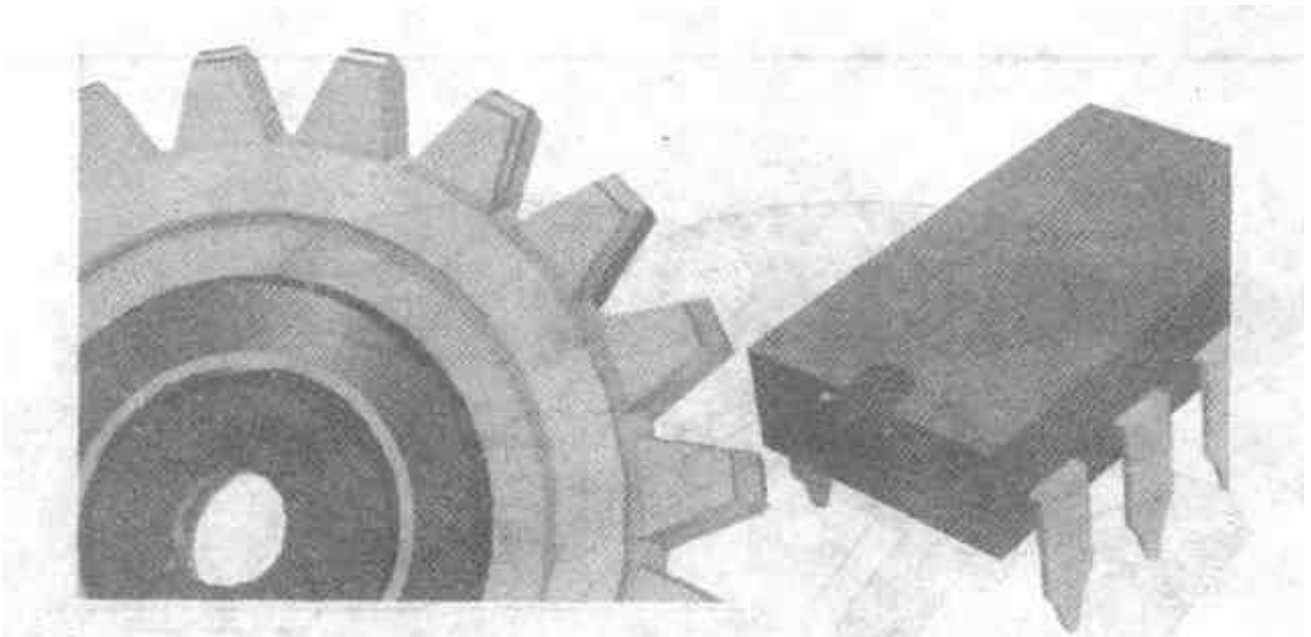
第9章 相扑机器人的制作技巧 210

1 相扑机器人的定义 …… 211
2 相扑机器人的体检 …… 214
3 相扑机器人的组装(1) …… 217
4 相扑机器人的组装(2) …… 220
5 相扑机器人的组装(3) …… 223
6 相扑机器人的组装(4) …… 226
7 相扑机器人的控制 …… 229
8 相扑机器人比赛大会规则摘要 …… 232



第1章

机电一体化的控制装置



机电一体化控制装置的发展

20世纪50年代，从洗衣机、黑白电视机进入家庭开始，兴起了工厂中的自动化风潮，运筹学与质量控制这些名词开始广泛传播。

20世纪60年代，日本进入以“日本列岛改造论”为特征的工业成长时期，开始了以奥林匹克为标志的繁华时代，随着高速公路、新干线的建成运营，街道上的汽车开始拥挤起来。号称3C的三大件，汽车（Car）、彩色电视机（ColourTV）与电冰箱（Cooler）已开始进入家庭。

20世纪70年代，在与电气设备相关的产业中，要求自动化、省力化的呼声已越来越高，控制装置被装入了产业机械之中，从而使劳动生产率得到飞速提高。

另外，由于半导体技术的快速发展和大规模集成电路的出现，进而进入微电子技术时代，也就进入了机电一体化的时代。机电一体化是传统的机械装置要素与电子装置要素的复合体作为特征的控制装置，现在不仅用在汽车与家用电器上，也用到了工厂里的生产设备上。

由于控制装置中装入了计算机而推进的机电一体化，使省力化、自动化更快地发展，制造出更方便、更小巧、更可靠、更具智能化功能的产品。

若以印制电路板用继电器的产量来考察这一机电一体化进展程度的话，1985年度的年产量为4亿个，八年后1993年度的年产量为10亿个。10亿这个数字就意味着每个日本人拥有10个当年所生产的印制电路板用的继电器。由此数字可推测，社会生活中所用的控制装置数量是多么巨大。

1 机电一体化的支持技术

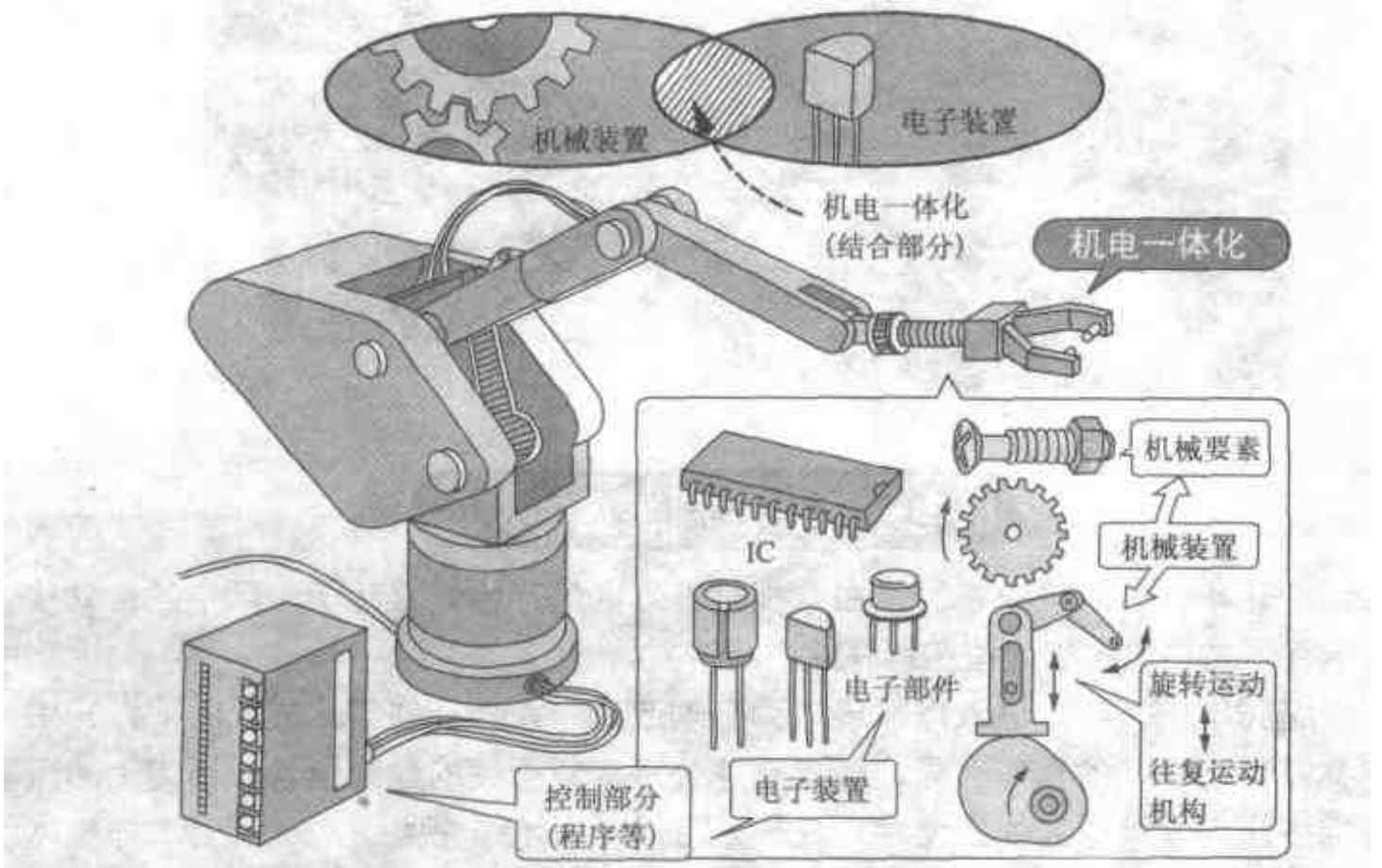


图 1　机电一体化示例

Ⅰ 机电一体化的定义

“机电一体化”这个名词，现在已作为极普通的术语，在电视或新闻等有关大众传播媒体中广泛使用。

这一广为传播的术语，是机械意义上的机械技术与电气、电子意义上的电子学技术相结合，从而产生的日本式英语“机电一体化”(mechatronics)。

图 1 所示的机械手，可作为机电一体化的典型例子。其机械部分由螺钉、

齿轮、弹簧等机械零件和把它们组合起来的连杆机构组成。而作为信息处理装置的电子装置部分，为了得到更好的控制性能，由集成电路、电阻与电容器等电子电路元器件构成。

机电一体化这一术语产生的背景，可归结为 1980 年以来电子技术、信息技术的飞速进步以及这些技术与机械技术优势互补的结果。

现在，依靠机电一体化技术，用更新的技术进行设计、制造与开发，就能创造出高度机电一体化的机器。这就是所说的“机电一体化的发展，开创了今天的高度信息化时代”。

Ⅱ 机电一体化的背景

我们考察一下机电一体化成为现代产业发展中心的背景。

大规模集成电路性能的提高与廉价销售

从真空管发展到晶体管是一个很大的变革，其特征可列举如下：

① 体积缩小；

② 性能的提高和功耗的降低；

③ 廉价销售。

从晶体管发展到大规模集成电路，再进一步发展到超大规模集成电路，更加快并强化了上述的变革特征。使机电一体化以更快地高速度向前迈进。

微处理器(超小型运算处理装置)技术的发展

随着超大规模集成电路技术的发展，开发出了使用更高性能微处理器的计算机。以前要用大型计算机处理的工作，现在用小型化后很容易装入机器中的计算机，同样也能处理。

传感器的发展与信息的利用

举例来说，像图 2 所示的液面控制装置，现在还是有继续使用浮子进行沉浮控制，以实现阀门的打开与关闭操作的。

现在，可利用半导体传感器对液面进行控制，只要把随时间变化液而高度、变化幅度等物理量信息，变换为电信号提取出来，就能按要求进行控制。因此，由于传感器的发展，使得迄今为止作为信息处理有困难的化学量等，也能把它转换为数字量来处理。

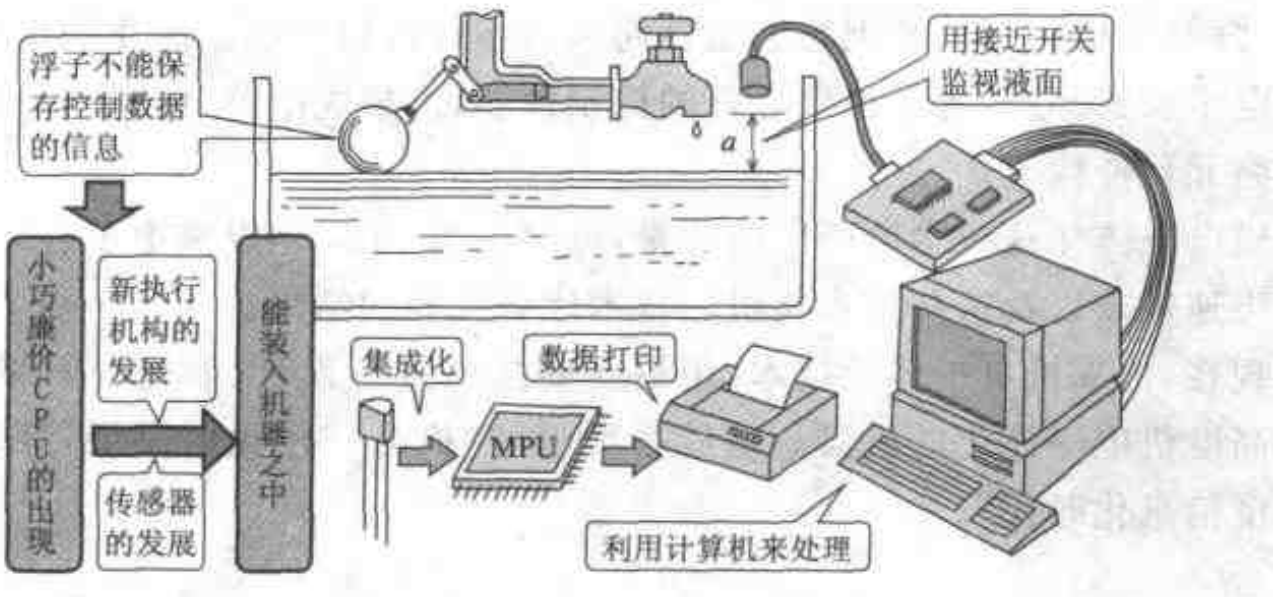


图 2　液面控制

执行机构的开发

在机电一体化发展的同时，也开发了便于用电信号控制的驱动装置，例如步进电机等新的执行机构产品。借助这种具有机电一体化功能的执行机构，能开发出以前的机器或设备上无法实现的高精确度、高精密度的工件制品。同时，要想制造满足消费者需要的新式产品，只要通过变更控制程序，就能尽快且方便地实现。

在家庭生活中，也已充满了内置微型计算机的电气设备。所以说，机电一体化的发展会对现代社会带来无法估量的巨大影响。

2 常见的机电一体化

为了更好地理解机械（设备）的动作原理，现以平面绘图仪为例，来考察一下机电一体化的构造。

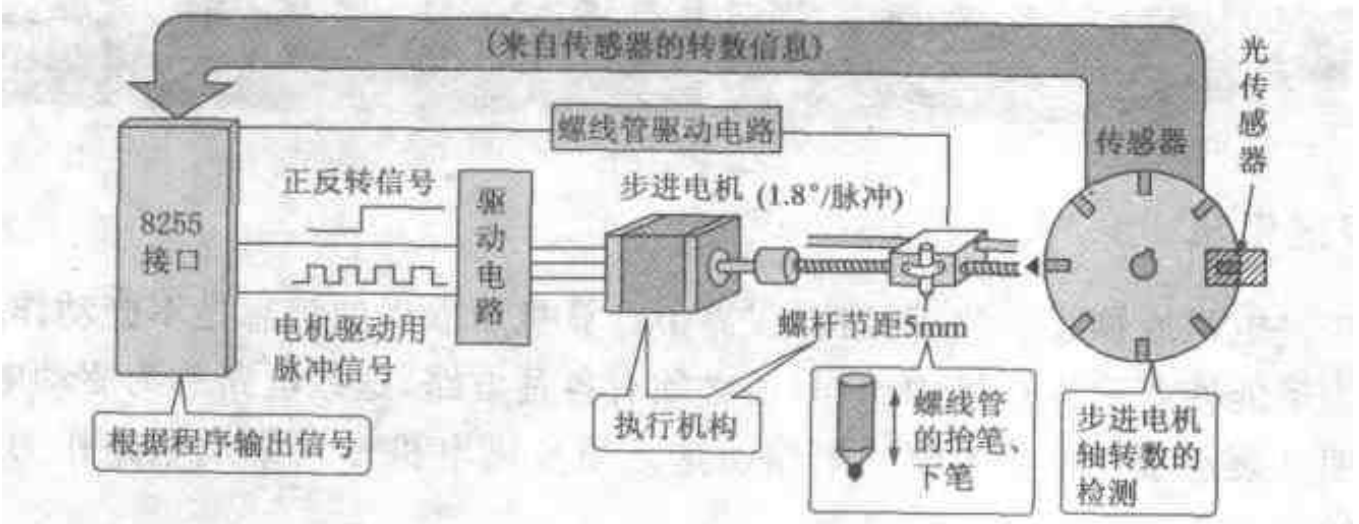


图 1 平面绘图仪的系统

I 平面绘图仪的系统

先考察一下关于平面绘图仪运动执行机构的控制系统。

步进电机的驱动

根据从图 1 所示的计算机中取得的信号，X 方向的步进电机或 Y 方向的步进电机会转动一个相应的角度。

画笔的抬起与放下(螺线管)

图 1 所示的绘图仪中，绘图与否是通过螺线管使画笔放下与抬起来实现的。步进电机与螺线管，就相当于平面绘图仪的两种执行机构。

II　传感器输入

转数(转角)的控制

图 1 所示平面绘图仪处于未使用状态，图中所示的步进电机驱动轴的一端，装有带缝隙的转盘，若装上穿透型光传感器，就能检测出轴的转数(即步进电机的转数)。以此对控制的状态进行监视，并对产生的状态进行控制，就称为闭环控制。与此对照，按照程序，按照指令的顺序来进行控制的方式，称为开环控制。一般来说，闭环控制能实现的控制精度高。

III　驱动电路

步进电机

步进电机或螺线管执行机构，仅靠由计算机所取得的信息是不能动作的。为了能使各种执行机构动作，还必须有各种电路。这些电路称为驱动电路。

图 2 表示步进电机的驱动电路。这是用步进电机专用集成电路作为驱动电路的例子。

确定步进电机正、反转转向用的信号，与使步进电机旋转用的脉冲信号，都必须有相应的输入端子。另外，对步进电机施加励磁有三种方式可供选择，即单相励磁、双相励磁与单、双相混合励磁。为此，还必须备有方式选择用的端子。

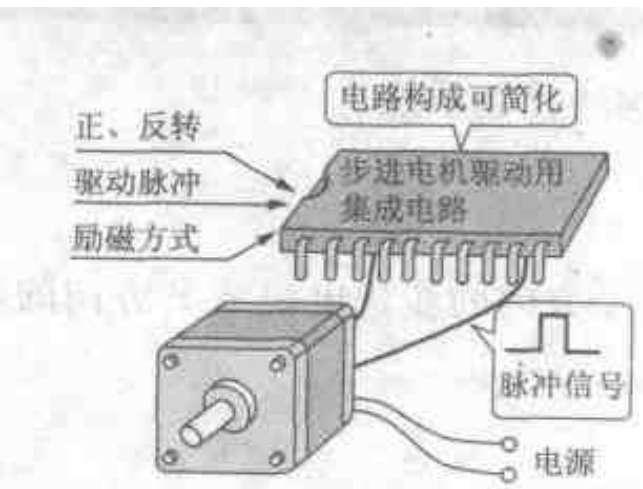


图 2　步进电机驱动电路

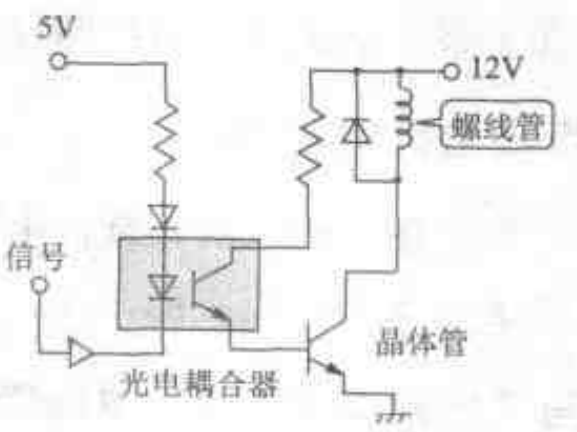


图 3　螺线管驱动电路

如图 2 所示，若用各种不同的步进电机专用集成电路作为驱动电路，则可简化电路结构，并能有效地发挥执行机构的功能。

螺线管

图 3 是放下与抬起画笔用的螺线管驱动电路的例子。取自接口的信号，经过晶体管的放大，便可得到驱动螺线管所必须的功率。

另外，像螺线管这类执行机构，在断开励磁时，会产生浪涌电压与噪声干扰等现象，并通过接口传入计算机，成为产生控制误动作的起因，可用光电耦合器来消除这种影响。

根除这种噪声干扰产生的误动作，是机电一体化中至关重要的课题。

Ⅳ 工作机构

画笔移动机构

画笔的移动，是通过连轴器把步进电机的旋转传递给螺杆位移机构来实现的。绘图仪的精度则由相当于步进电机一个脉冲转角所传递的导程来决定。

这样，把工作机构与执行机构的作用联系在一起，就能实现一种功能的机电一体化。

画笔的连杆部分

图 4 所示为利用弹簧的弹力机构使螺线管复位。

综上可见，机电一体化就是把电气、电子技术与机械技术结合起来的技术。

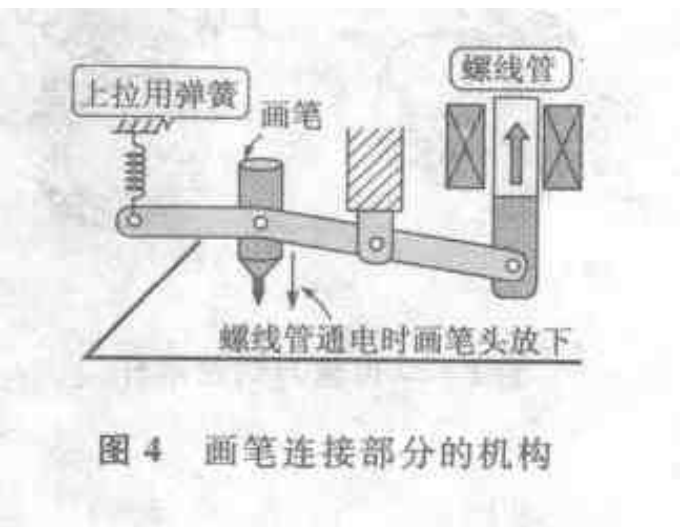


图 4 画笔连接部分的机构

3 机电一体化的机械装置



在机电一体化中，利用电动机的旋转运动来驱动机械装置，构成这类机械装置的部件称为“机械零件”。机械零件中传递旋转运动常用的有齿轮与固定部件用(将一些部件固定)的螺丝。这里介绍它们的基础知识。

1 齿轮的基础知识

齿轮传动的特征

齿轮能可靠地传递电动机等的旋转运动。通过改变齿轮的齿数，可自由地改变转速比。根据齿轮的种类，有的齿轮能传递齿轮轴的角度。

[例]　伞齿轮等的利用(参见图 1-(a))。

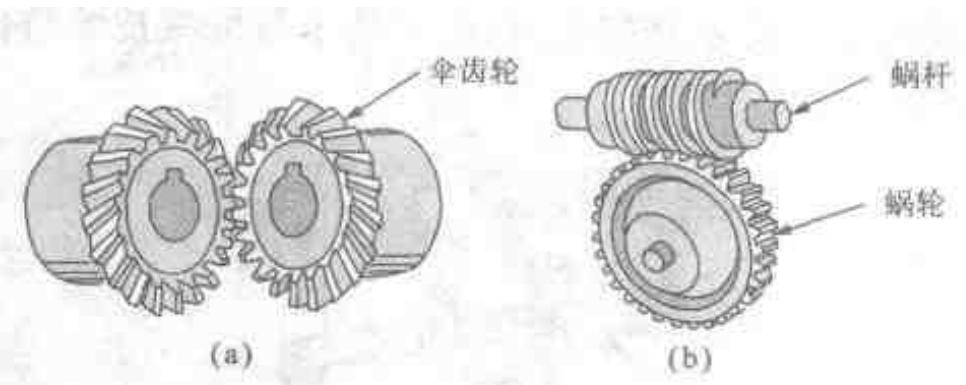


图 1　伞齿轮与蜗轮蜗杆

有的齿轮能传递大的减速比。

[例]　蜗杆与蜗轮的利用(参见图 1-(b))。

齿的大小

齿越大则能传递的力越大。常用的渐开线齿轮，用“m＝模数”来表示齿的大小。实际齿的大小(高度)是此模数的 2.25 倍。如图 2 所示，若齿的高度约为 9mm，则 9/2.25＝4，所以齿轮的齿的大小为模数 m＝4 时的大小。

由齿轮外径求模数的方法

① 用尺或游标卡尺测量齿轮外径。

② 数该齿轮的齿数。

③ 代入下列算式求 m。

m＝外径/(齿数＋2)。



图 2　模数(齿的大小)

轴距(轴与轴的中心之间距离)的求法

由齿轮的模数(m)与两个齿轮的齿数(Z_1、Z_2)，可按下式算出两个齿轮啮合时的轴距：

$$轴距＝m(Z_1＋Z_2)/2$$

例　题

① 模数求法

有一个外径为 64mm 的齿轮(参见图 3)，求齿数为 Z_1＝30 个时的模数。可以按下式求得：

$$m＝64/(30＋2)$$

得到 m＝2。

② 齿数比(减速比)(参见图 4)

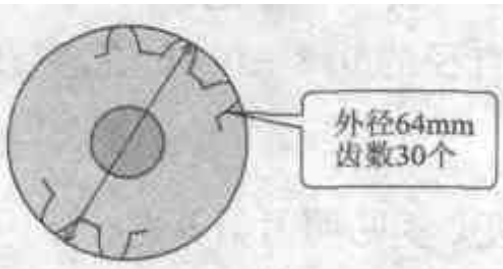


图 3　齿轮模数的计算例子

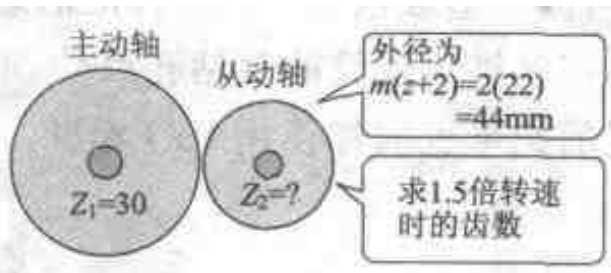


图 4　减速比

Z_1 为主动轴齿轮的齿数，要使从动轴以 1.5 倍的速度旋转，则从动轴的齿轮数 Z_2 可按下列计算求得。

由齿数比 i 为 $1.5＝Z_1/Z_2$ 得

$$Z_2＝Z_1/1.5$$

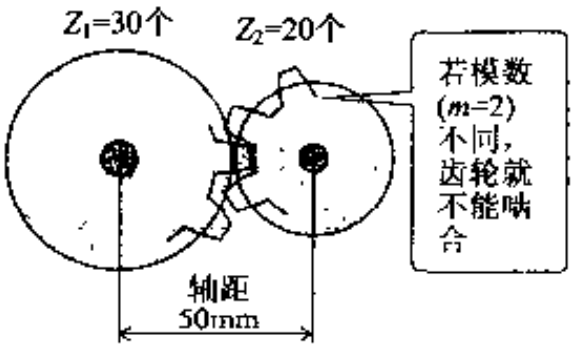


图 5　轴　距

代入 $Z_1=30$，则

$$Z_2=30/1.5=20$$

所以 Z_2 的齿数为 20 个。

③ 轴距(参见图 5)

齿数 $Z_1=30$ 个与 $Z_2=20$ 个的齿轮啮合时的轴间距离，按下式计算：

$$\text{轴距}=m(Z_1+Z_2)/2=2(30+20)/2=50$$

Ⅱ　螺　纹

三角形螺纹

螺纹的规格要符合 ISO(国际标准化组织)标准，以保证互换性。

两个部件(机械零件)之间的固定，主要用三角形螺纹(公制普通螺纹)。图 6 示出了三角螺纹各部分的名称。

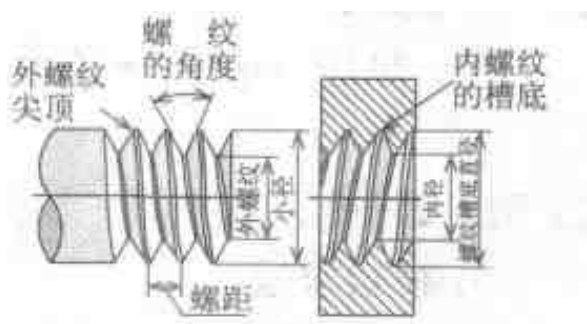


图 6　螺纹的名称

螺纹标准与否，用下列标准化尺寸来衡量：

① 螺纹外径(称为公称尺寸)；

② 螺纹角度(60°)；

③ 螺距(从螺纹顶尖到螺纹顶尖的长度)。

螺纹的标准因在国际上尚未统一，现在还常出现各种各样的标准。在制造新的机电一体化装置时，应尽量使用 ISO 标准的螺纹，尽管库中有旧标准的存货，会因此造成浪费。ISO 标准的螺钉头上有“M”或“·”标记。

用丝锥攻螺纹的预钻孔尺寸，取螺钉公称直径的 0.8～0.85 倍。与螺钉相对应的螺母也同样使用 ISO 标准。

滚珠丝杠(参见照片 1)



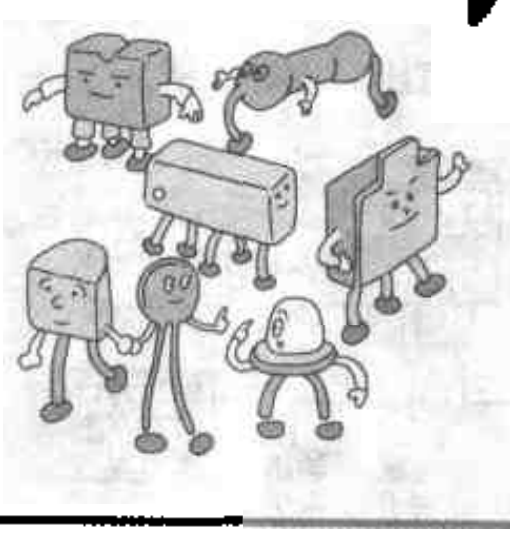
照片 1　滚珠丝杠

机电一体化在机床工作台的驱动之类应用中，在内螺纹与外螺纹之间加入滚珠，在进给螺纹中，滚珠丝杠起传递力的作用。

4 机电一体化的电路知识(1)

在学习和实践中，如想掌握机电一体化的基础理论或接口技术，都需要电气和电子的基础知识。

以下介绍有关实际常用元器件的标准及表示方法等基础知识。



Ⅰ 电　源

由于计算机或接口控制电路都要用集成电路(IC)，如晶体管-晶体管逻辑电路(TTL)或互补金属氧化物半导体(CMOS)电路，所以需要用直流 5V 作为驱动电源(参见图 1)。

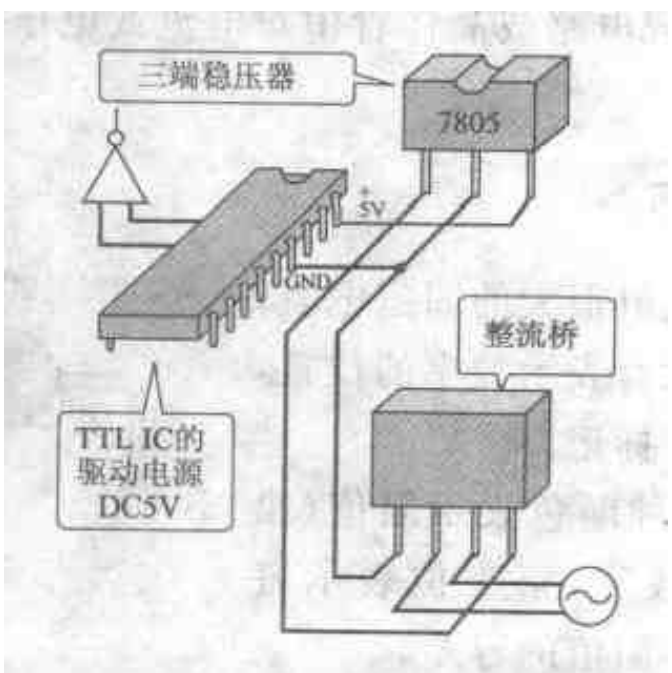


图 1　IC 用驱动电源

另外，运算放大器(OP)或数-模转换器(D/A)等，需要±12V、±18V 的电源。电源可用市购或自制的直流电源。电源容量应根据电路所需要的电流来选择。如果执行机构的驱动电源由其他电源供给，那么电源容量用 2A 左右就可以了。

II　三端稳压器

为了稳压，用图 2 所示的三端稳压器。三端稳压器的 4 位数字定义如下：

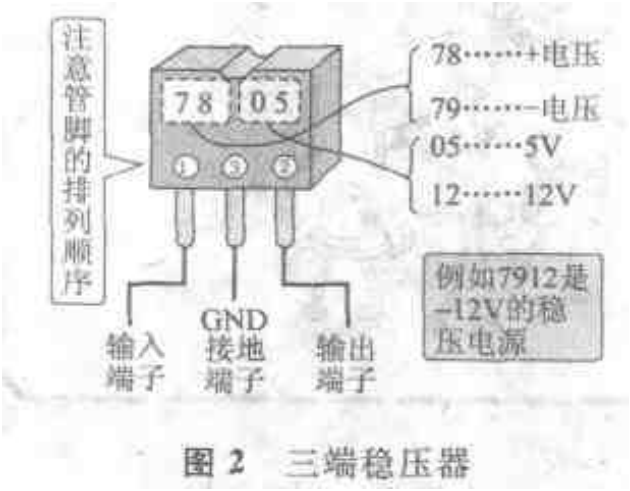


图 2　三端稳压器

头两位数字是相对地(GND)的输出电压的正或负；后两位数表示固定电压的大小。例如：

7805 为＋5V　7905 为－5V

7812 为＋12V　7912 为－12V

三端稳压器的输入端所加的电压，要比输出端输出的电压高一些。输出端电压为 5V，输入电压可为 7～18V 左右。

把这一输入电压与输出电压的能量之差以热量的形式散发出去，以实现稳压，所以三端稳压器必须有散热片。

III　电　阻

电阻也称为电阻器，顾名思义，电阻是阻止电路中电流流动的元器件。利用这一性质，可用于集成电路或晶体管中对电流或电压进行必要的调整，以使器件正常工作。

固定电阻器的色环

电子电路都用电阻值很大的固定电阻，其中小型固定电阻因没有表示数字的位置，阻值及其容差用色环来标记。

如图 3 所示，前三种颜色表示阻值(第一、第二位表示有效数字，第三位表示幂次)，最后一种颜色表示阻值的容差。

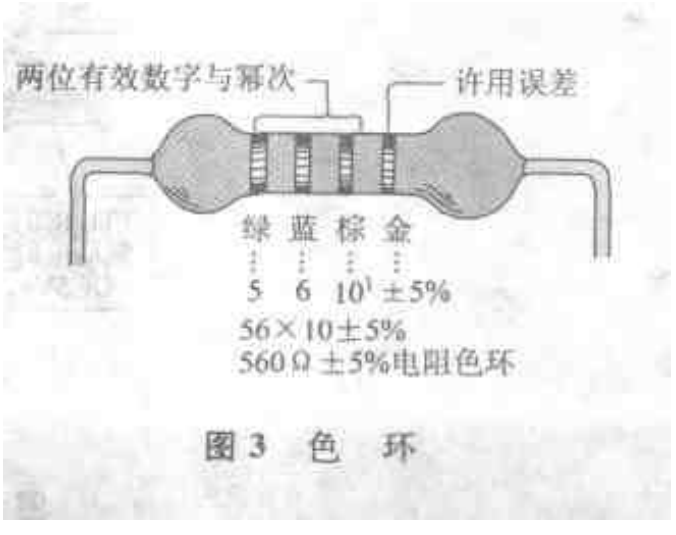


图 3　色　环

额定功率

选择电阻时，尽管常常忽视额定功率，但因流过电阻的电流会产生热量，额定功率是它很重要的指标，通常用的电阻额定功率为 0.125～5W 左右。

可按图 4 求得固定电阻器的额定功率，应满足以下关系：

电阻器的额定功率＞电阻器的消耗功率

实际上，要取 2～2.5 倍的冗余量。

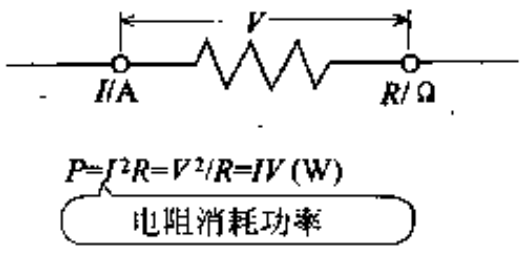


图 4　电阻器的消耗功率

可变电阻

可变电阻因阻值可变化，一般以最大电阻值表示。

电位器：通常使用转角与阻值成线性关系的电阻器。

半可调电阻：用于调整一次后几乎不需要再变更阻值的场合。

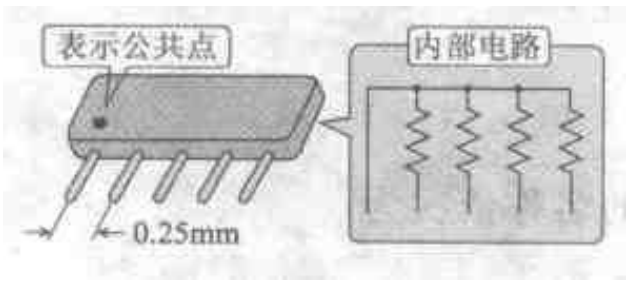


图 5　电阻阵列

电阻阵列(集成电阻)

是多个电阻器封装为一体的电阻器，它有减少元器件个数，有效利用主板面积的优点。除了图 5 所示的带公共端(公用)的 4 元件型外，还有 8 元件型的也常用。端子的间隔与 IC 引脚间距(0.25mm)相适应，所以能够组装到基片上。

电阻的色环表示

色　名	第 1 色带	第 2 色带	第 3 色带	第 4 色带
	第 1 位数字	第 2 位数字	幂　次	标称电阻值允许误差
黑　色	0	0	10^0	—
棕　色	1	1	10^1	±1%
红　色	2	2	10^2	±2%
橙　色	3	3	10^3	—
黄　色	4	4	10^4	—
绿　色	5	5	10^5	±0.5%
蓝　色	6	6	10^6	—
紫　色	7	7	10^7	—
灰　色	8	8	10^8	—
白　色	9	9	10^9	—
金　色	—	—	10^{-1}	±5%
银　色	—	—	10^{-2}	±10%
—	—	—	—	±20%

5 机电一体化的电路知识(2)

1 晶体管

晶体管的形状

晶体管具有如图 1 所示的三个管脚，管脚的名称依次为

发射极　E
集电极　C
基　极　B

图 1 示出了两种类型(PNP 型、NPN 型)的晶体管。通常管脚排列如图 1 所示，但也有例外的情况，所以在使用时，最好查阅技术参数表。

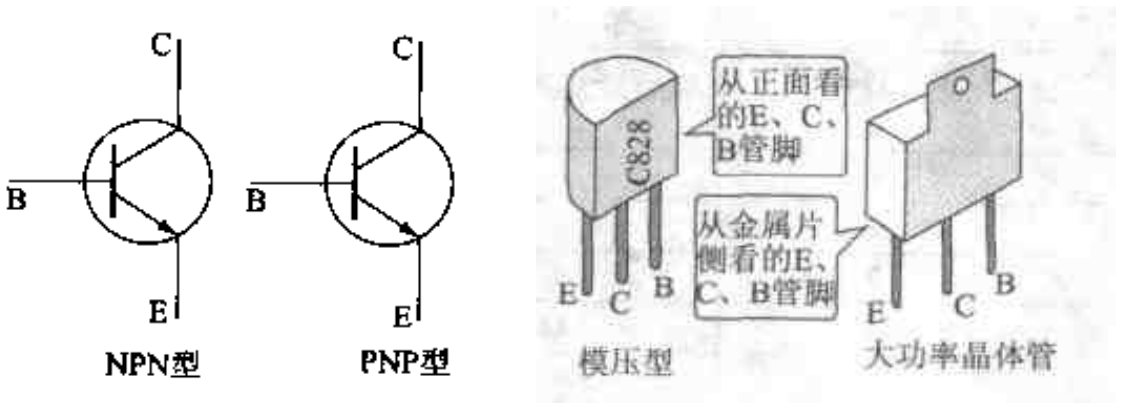


图 1　晶体管的形状

晶体管管脚辨别方法

晶体管可看作是由两个二极管连接成的器件，可进行测试，如图 2 所示。通过确定二极管的正向，即可辨别晶体管管脚，需要的话，也能检查晶体管的好坏。

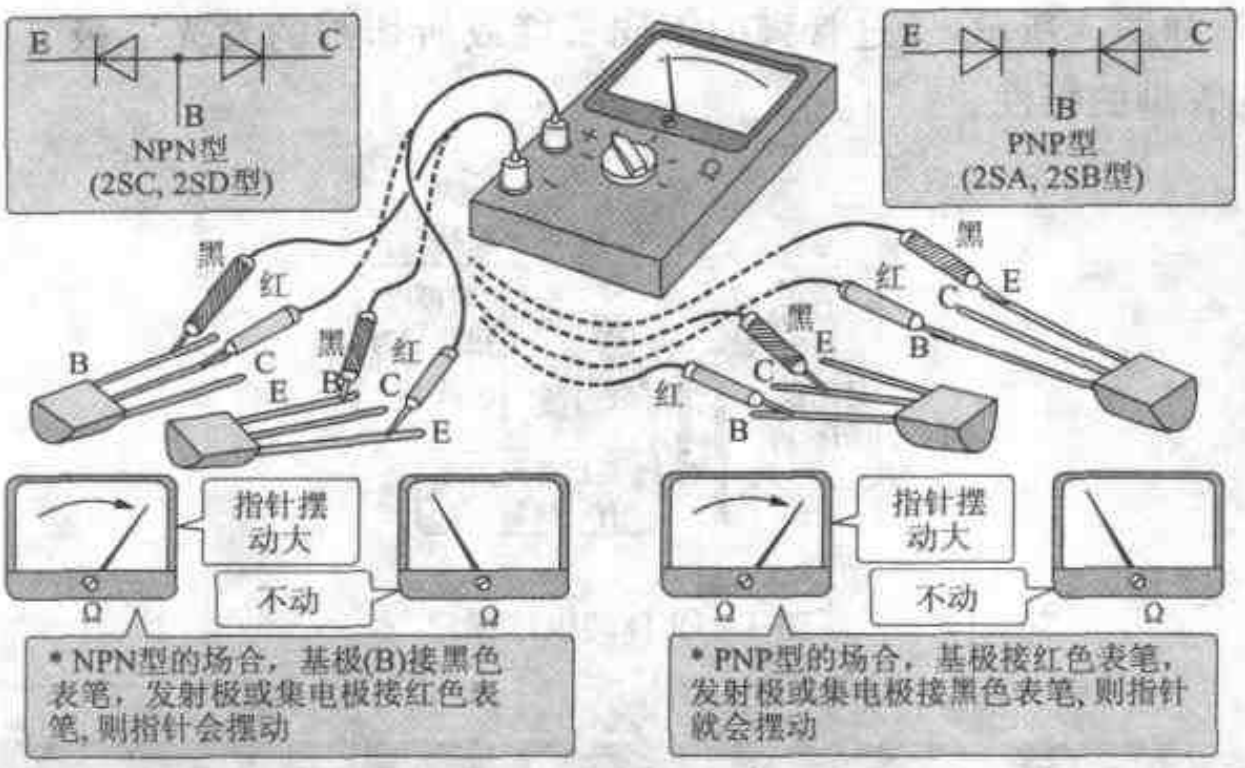


图 2 晶体管管脚辨别方法

Ⅱ 二极管

二极管有两个管脚，有单方向通过电流的整流作用。在施加正向电压时，电流通过的方向为正向(正极到负极的方向)，把不通电流的管脚方向称为反向(负极到正极的方向)。要辨别是哪个管脚时，可用万用表按图 3 方法测试。

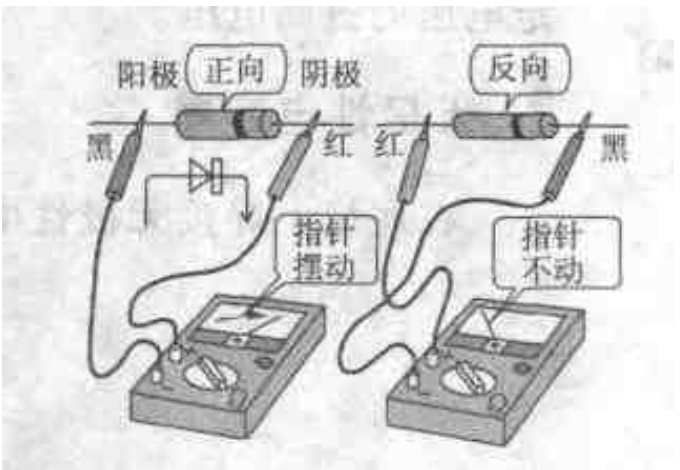


图 3 二极管的测试

Ⅲ 发光二极管

发光二极管(LED)具有正向通过 10mA 左右的电流时就发光的性质。发光二极管的测试方法基本上与图 3 相同。万用表指针摆动，则黑色表笔所接的就是正极。

通常，如图 4 所示，通过管脚的不同长度或所切割的发光二极管管体形状也能辨别管脚的极性。

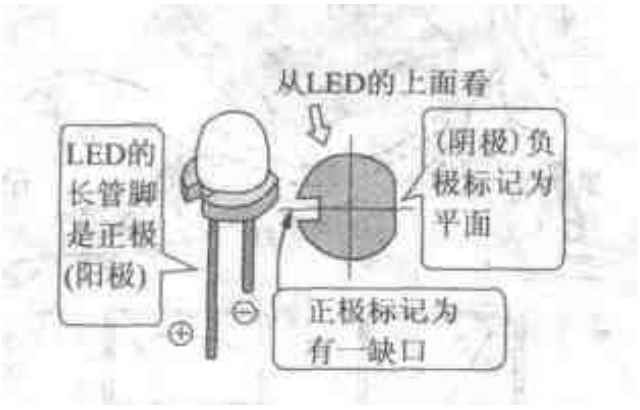


图 4　LED 的极性标记

Ⅳ 电容器

可以从有无正负极性分为有极性电容器(电解电容器)和无极性电容器。

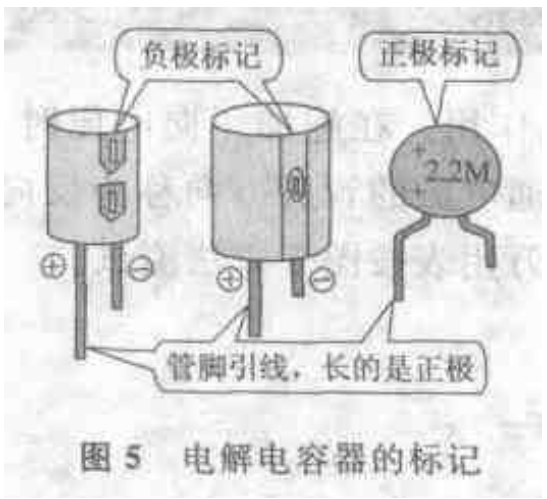


图 5　电解电容器的标记

有极性电容器

如图 5 所示，有极性电容器以不同的管脚长度或“－”标记来表示极性。在电路中安装电容器时，必须十分注意其极性。并且不得施加大于额定电压的过高电压。

无极性电容器

在底板上焊接无极性电容器时，不必区别其管脚的极性。

电容器容量的表示

电容器容量的表示如图 6 所示。三位数字中，头两位是常数，接着的数字为 10 的幂次数。所得到的数值单位为 pF(皮法)。

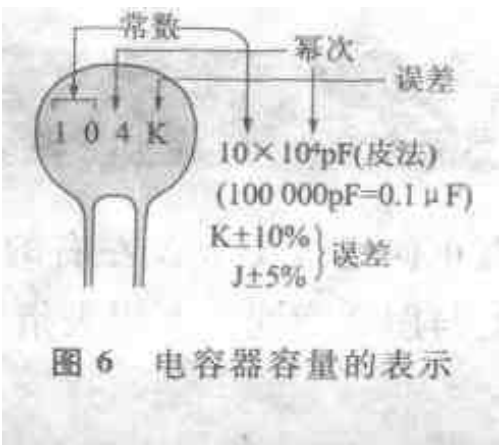
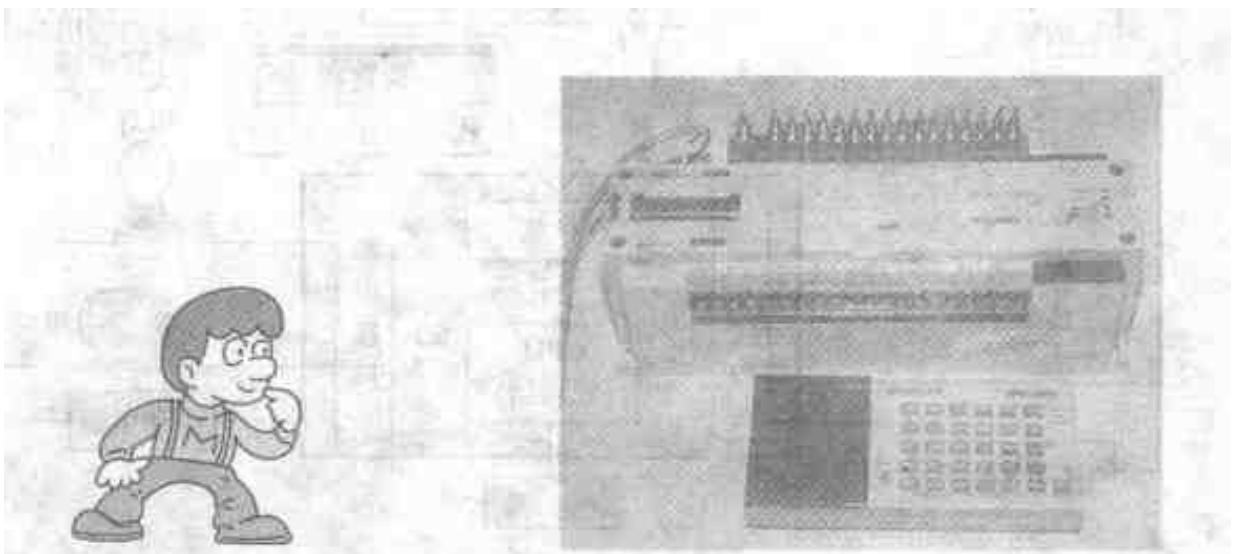


图 6　电容器容量的表示

6 可编程序控制器



照片 1　PC 的外观

I　可编程序控制器的定义

生产现场希望能结合市场需要“多品种、少数量”地生产。

生产现场需要自动化(FA:Factory Automation)生产系统的同时,也需要能适应多种产品的柔性制造系统(FMS:Flexible Manufacturing System)。这种生产方式,通常使用一种顺序控制器即 PC(可编程序控制器)。

在生产现场中的数控机床或加工中心,机器人的控制或管理上,PC 均起重要的作用。并且,在电梯等自动化系统中,也用 PC 作为控制器。

PC 是在 1969 年开发的,以后随着半导体、微处理器技术的发展,成了能使用程序的多功能控制器。可以说 PC 就是生产现场用的机电一体化装置。照片 1 为 PC 的外观。

Ⅱ　PC 的结构

如图 1 所示，PC 由输入接口、输出接口、存储器及 CPU 构成。通过相当于个人计算机键盘的编程板，可对 PC 内部的存储器编程序，并使其按程序进行控制。

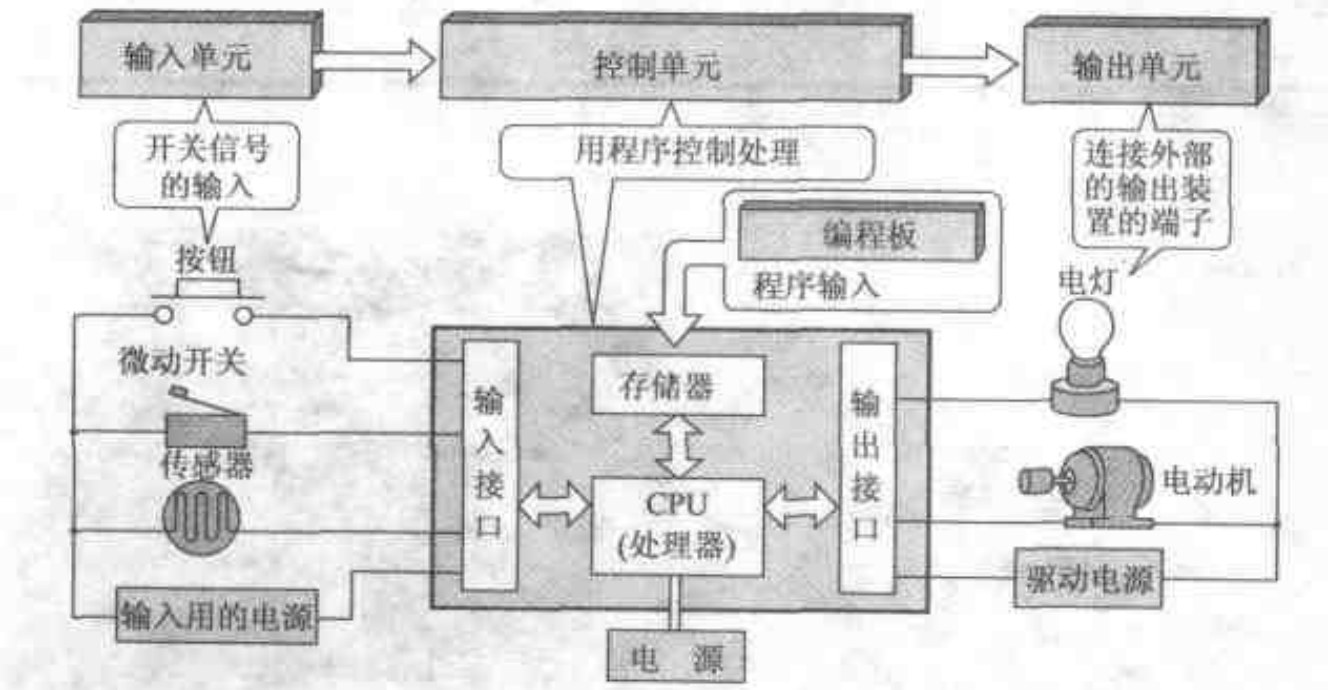


图 1　PC 的结构

Ⅲ　PC 内的信号流

PC 内的信号流示于图 2，根据从外部取得的开关等的信号，使 PC 内部的输入继电器接点动作。由此信号，按 PC 本体内存储器中所存的事先编好的程序，进行 PC 内部的控制。其结果，使 PC 内的输出继电器动作。

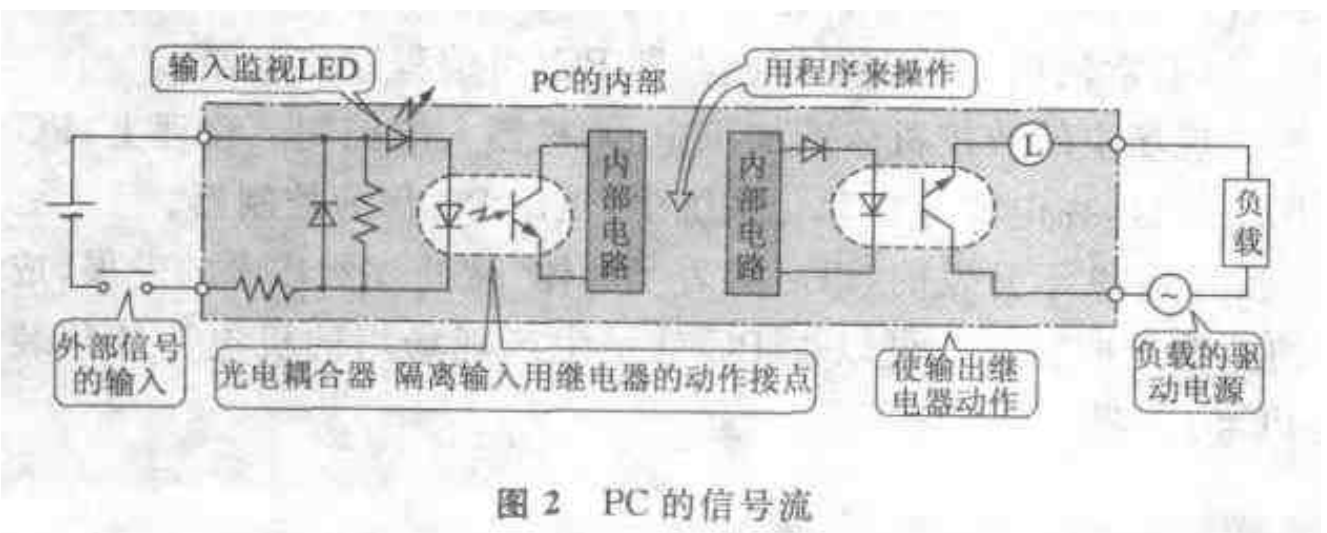


图 2　PC 的信号流

通过这一系列的动作，对执行机构进行控制。

Ⅳ PC的特征

与迄今为止生产的现场常用的有接点继电器控制对照，对有关PC(可编程序控制器)的无接点控制的特征进行探讨。

① **控制功能** PC与有接点继电器的控制不同，只按程序对接在PC本体继电器端子上的输入装置、输出装置进行控制，因此要变更其控制内容时，不必改动接线，只要变更程序就可以改变控制内容。

② **可靠性** PC是由半导体器件所组成的，不存在有接点继电器的接触不良的问题。

③ **维护** PC可以以控制单元为单位进行更换，因此维护方便。

④ **主体结构** PC装置体积小，重量轻，具有极好的环境承受能力。

⑤ **简易性** 与个人计算机程序相比，PC的程序编制要简单一些。

⑥ **保存** 能保存PC的程序，用于其他控制时，只需简单地变更程序，配置外围装置。

⑦ **系统化** 可把任意个PC组成一个系统，并且可用计算机对PC进行集中管理。

⑧ **缩短试验时间** 新安装的或更动的控制装置，都必须试运行，由于调整等很简单，可以缩短试验时间。

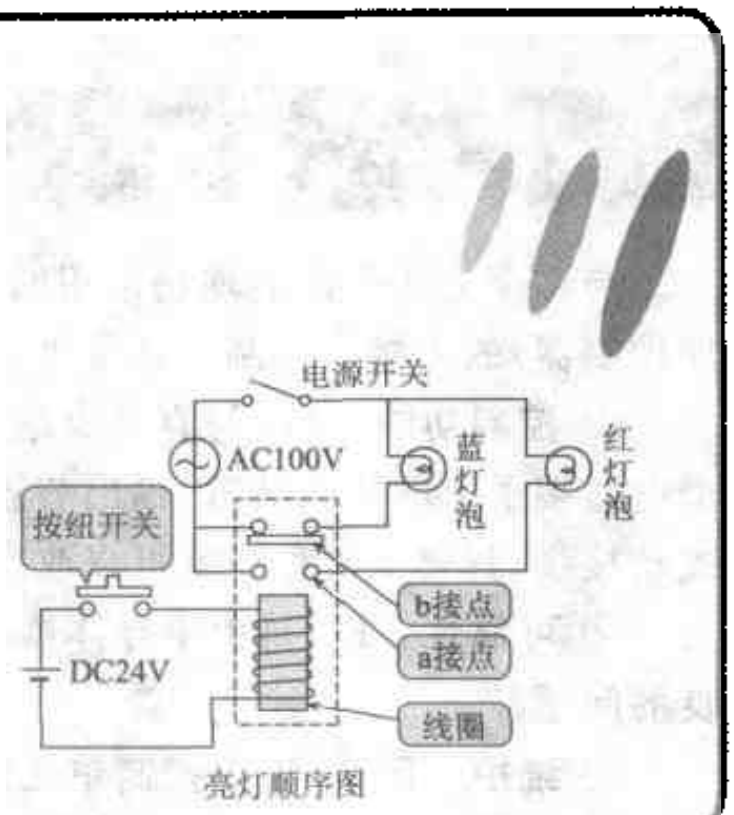
⑨ **重复使用** PC可以重复使用，即使作为控制对象的机械已报废，通过变更程序，仍然可以再将PC本体用于其他的控制机械中。

* * *

PC有上述优点，与黑匣子化的PC软件相比，继电器式有接点的控制，至今已有很丰富的专用技术，有接点控制具有硬件简单、易于理解的优点。

7 PC的编程

可用传统的顺序图来表现继电器或操作开关等硬件，例如右图，电源一接通，蓝色灯泡就亮，按钮一接通，红色灯泡就亮的顺序图等。



亮灯顺序图

Ⅰ PC用的梯形图

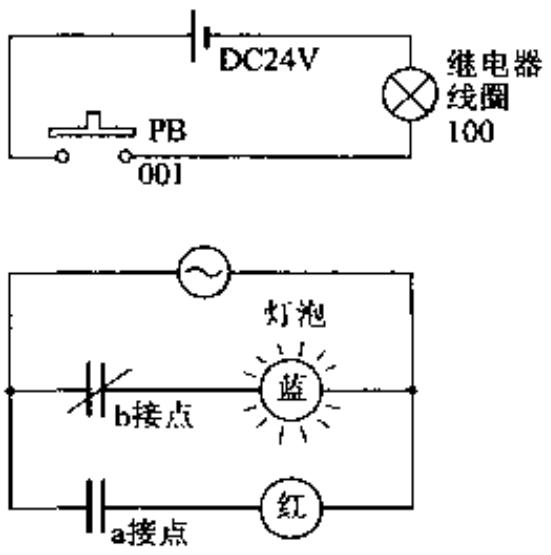


图 1　梯形图

PC 常用的是图 1 所示的控制流，用继电器符号表示的电路图（梯形图，Latter 图），Latter 就是梯子的意思，由于看起来像梯子，所以用这一术语。

为了将控制的内容装入 PC 的存储器中，要用 PC 的指令，这一作业称为编程。也就是说，将顺序图（梯形图）中所画的电路的接点及其连接方法，变换为指令语言，就成为所说的 PC 程序。

Ⅱ PC的指令

基本的 PC 指令由步（地址）、操作码、操作数构成。介绍一个常用的 PC 指令。

［例］　接在母线上的接点（参见图 2）。

LD：a 接点连接到母线的指令

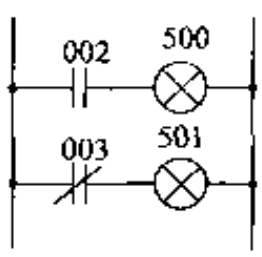
LD NOT：b 接点连接到母线的指令

OUT：驱动线圈的指令

[例] 串联、并联的连接接点(参见图 3)。

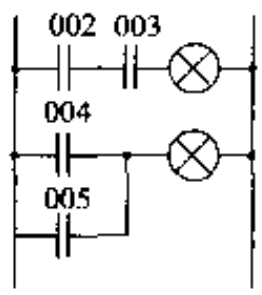
AND：a 接点串联连接的指令

OR：a 接点并联连接的指令



步	指令	数据
1	LD	002
2	OUT	500
3	LD NOT	003
4	OUT	501

图 2



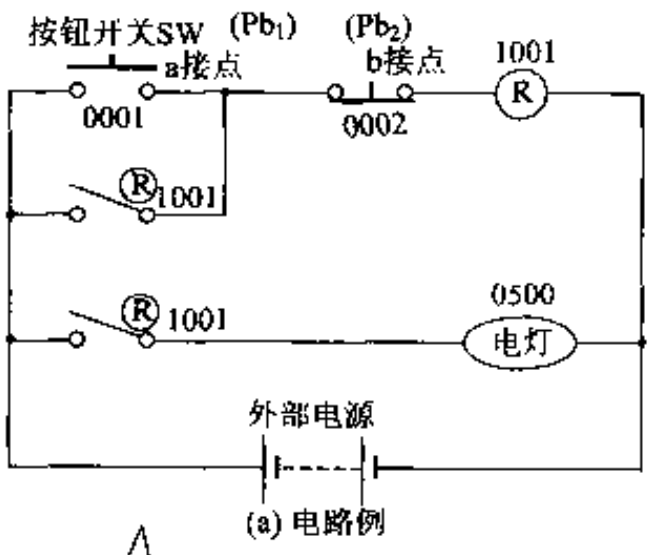
步	指令	数据
1	LD	002
2	AND	003
3	LD	004
4	OR	005

图 3

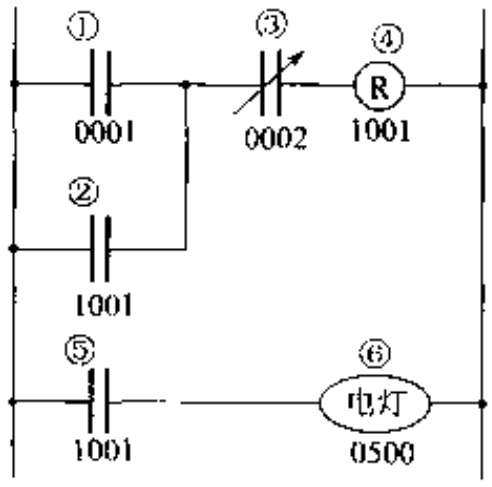
Ⅲ PC 程序举例

图 4(a)是按下按钮开关(Pb_1)时，PC 内部继电器(R)动作，成为亮灯的电路。该程序可使开关状态自保，按下 b 接点的按钮开关(Pb_2)，它的常闭接点就使灯熄灭。

图 4(b)为图(a)所示电路的梯形图。如把灯换成电动机，则该图就是电动机驱动控制图。



(a) 电路例



(b) (a)的梯形图

动作
①按下按钮开关(取得a接点信号0001)
②由1001输出信号，即使输入信号消失，也能自己保持，继续保持输出(灯亮)
③串联连接常闭接点。0002(AND NOT)上一输入信号，则1001的输出信号就停止(灯熄灭)

步	指令	数据
1	LD	0001
2	OR	1001
3	AND NOT	0002
4	OUT	1001
5	LD	1001
6	OUT	0500
7	END	

*输入END指令程序结束。

图 4

Ⅳ　PC 的使用方法

实际使用的 PC 控制程序，按图 5 所示，由分配输入输出装置与 PC 内部的继电器来组成梯形图，用 PC 指令编程，经修正，把程序存入只读存储器(ROM)后即可。

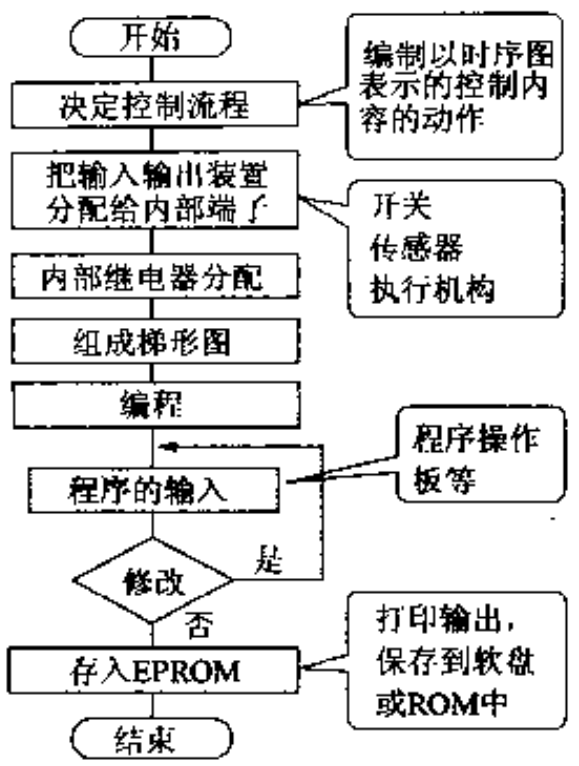


图 5　PC 的控制顺序

Q 挑战题

① 什么是机电一体化，它是由什么名词合成的？
② 产生机电一体化的背景是什么？
③ 什么是微处理器？
④ LSI 是什么的缩写？
⑤ 闭环控制是什么样的控制？
⑥ 列举两个以上使用机电一体化的机械零件。
⑦ 表示齿轮大小的单位是什么？
⑧ 什么是 ISO 螺纹？
⑨ 螺母的预钻孔直径大约是螺钉直径的多少倍？
⑩ 滚珠丝杠有何特征？
⑪ 提供直流电路中 +5V、+12V 等固定电压用的器件叫什么名称？
⑫ 试说明电阻的色环。
⑬ 列举一下晶体管的两种类型。
⑭ 什么是二极管电流的正方向？
⑮ 电容器表示的 104K 是什么意思？
⑯ PC 是什么的缩写？
⑰ 什么是梯形图？
⑱ 什么是 a 接点、b 接点？
⑲ 自保持电路是什么样的电路？

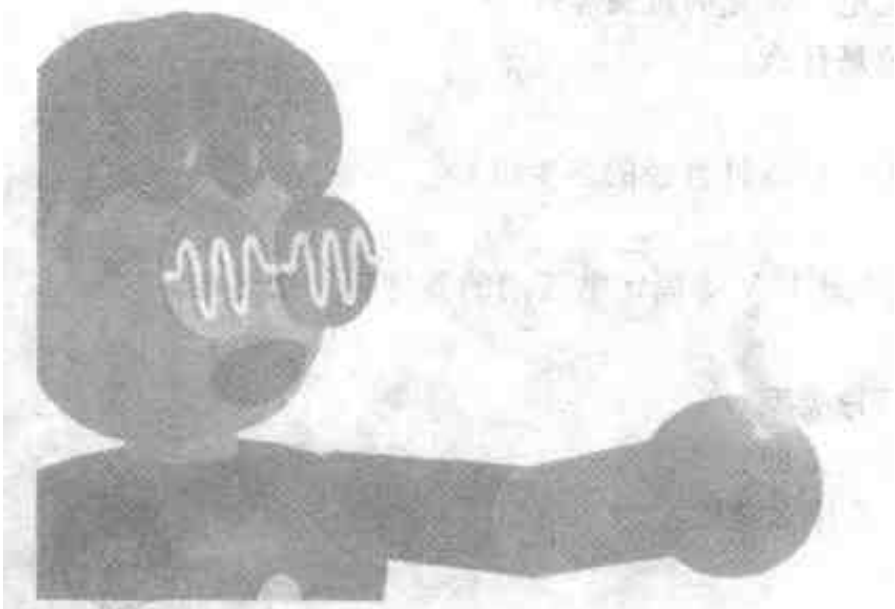


A 答 案

①机械学与电子学。 ②略，参见第 2 节。 ③MPU(超小型运算处理器)。 ④大规模集成电路。 ⑤测量被控制量，将其中一部分反馈与基准值比较并进行修正的控制方式。 ⑥齿轮、螺纹。 ⑦模数。 ⑧国际标准化组织的螺纹标准。 ⑨0.8～0.85 倍。 ⑩通过内螺纹与外螺纹之间的滚珠可平滑地传动。 ⑪三端稳压器。 ⑫略，参见第 5 节。 ⑬NPN 型，PNP 型。 ⑭从正极到负极的方向。 ⑮10×10^4 pF＝0.1μF。 ⑯可编程序控制器。 ⑰略，参见第 7 节。 ⑱a 接点：常闭接点；b 接点：常开接点。 ⑲由输入进入接通状态，并在此后保持接通常态的一种保持记忆的电路。

第2章

传感器的基础知识

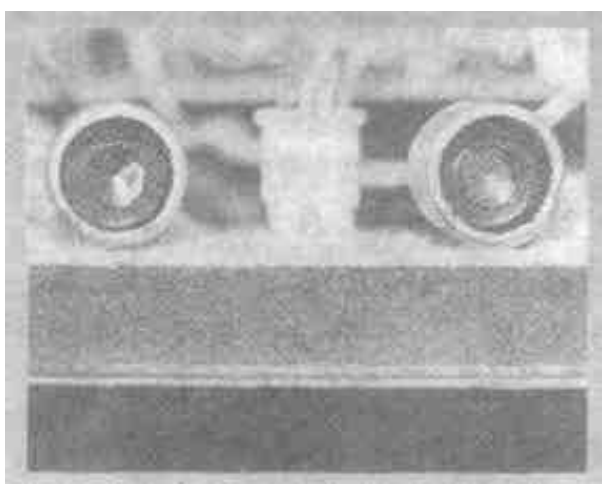


传感器的功能与信息检测、变换

英语“Sense”，作为“人的感受”时，是感觉的意思。另外也有“五官的感觉”的意思，如“闻到焦味”、“吃起来酸呀!”、“感到热”等等。一碰到沸腾着的水壶，就感到烫(信息的搜集=输入)，急忙把手缩回来(缩手行动=输出)，可以说明从得到感觉到行动成为一体。

由此例可知，我们的行动与“五官的感觉”有密切的关联。

人的五官(视觉、听觉、触觉、嗅觉、味觉)是用于搜集日常生活信息的，所以是很重要的器官。



照片P1 传感器

传感器(sensor)是与该词(sense)有关的术语，传感器可代替人的五官，在机电一体化的控制中，作为检测必要的信息并进行变换的装置。

要使机电一体化的执行机构动作也同样，依据明确指令规定动作，尤其必须有“来自传感器的信息”作为指令所需的信息依据。

这一章介绍有关用于机电一体化的传感器。

1 传感器的定义

前面已介绍过传感器相当于人的五官，在机电一体化中，传感器是“检测、变换电子机械起作用所需信息的眼睛”。与传感器类似的词叫“转换器”，是把力或速度之类的物理量转换为易于处理的输出信号的变换器，传感器也有同样的作用。

I 传感器的使用方法

传感器有信息检测与将检测到的信息转换为电气信号两大功能。关于传感器的作用，以图 1 所示的数控车床为例进行说明。

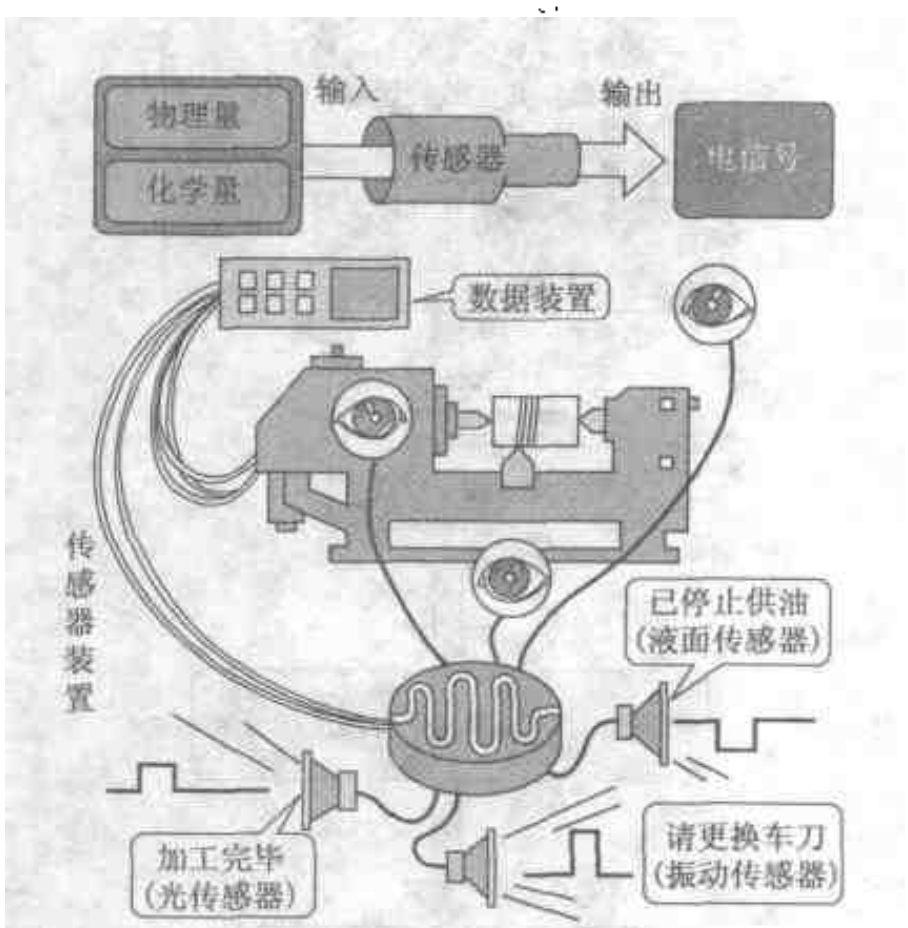


图 1 数控车床的传感器

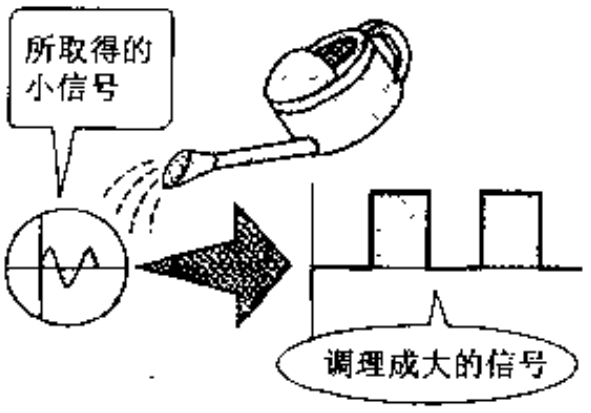


图 2　电信号元

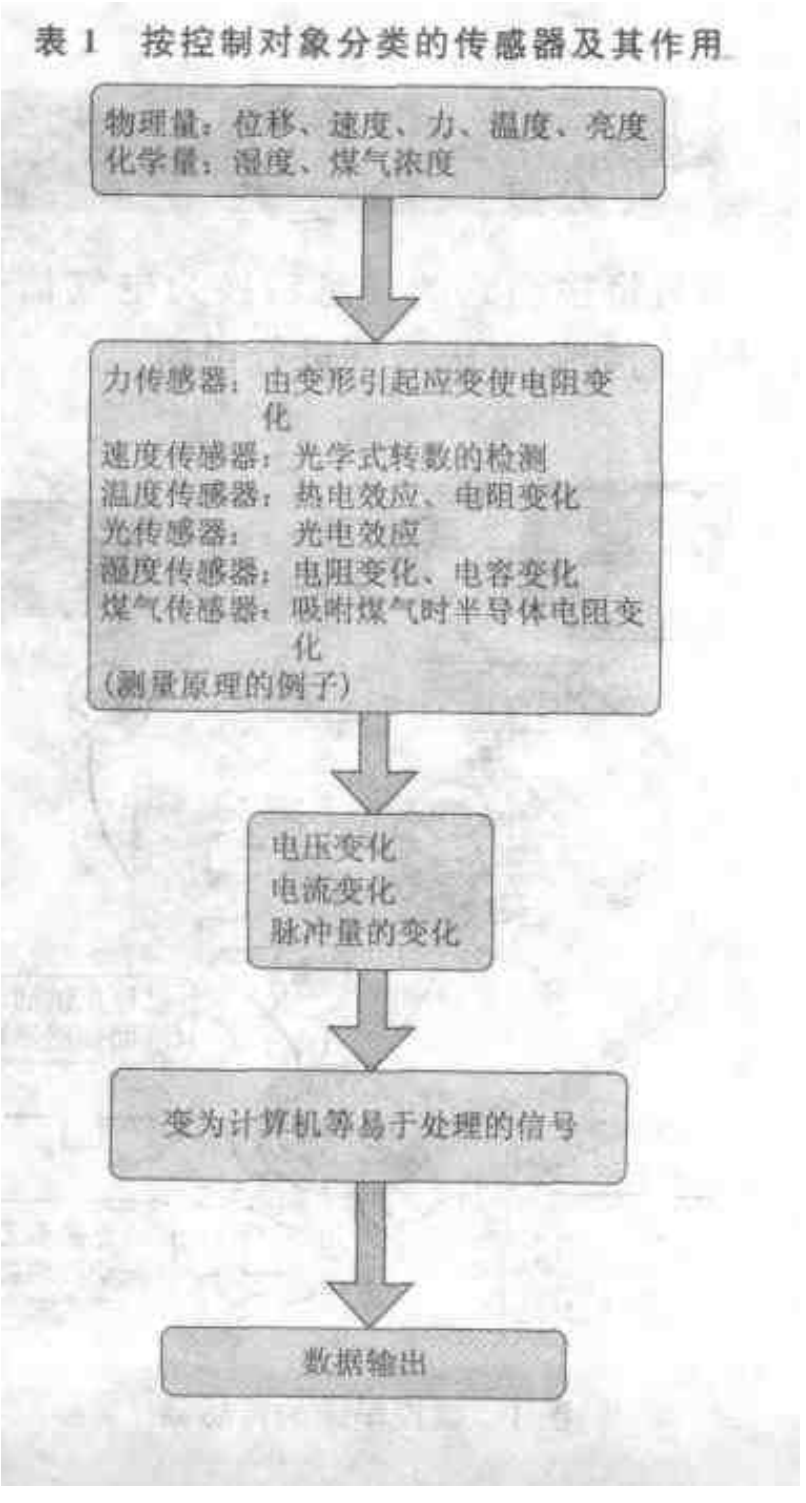
① 工作状态，在检测车刀位置的同时，用光传感器进行监视。

② 刀尖的形状对工作尺寸，表面精度等影响最大，用安装在车刀上的振动传感器检测车刀的磨钝程度，用振动的状态来监视。

③ 车床所用的润滑油，液压系统的油量，是能源的一部分，这种油量通过液面传感器进行监视。

这些传感器的信息，是作为电信号或电信号元(种子)来输出的。这一“电信号元”可用传感器检测出来，若要使它符合控制机械等对象的要求，需如图 2 所示的那样，将“信号元调理成更为合适的形状”。这一处理也称为“信号加工”，属于传感器使用技术的一部分。

表 1　按控制对象分类的传感器及其作用



Ⅱ 按控制对象分类的传感器

传感器的种类繁多，根据其使用的目的，可涉及到许多方面。就常用的来讲，有力、位移、速度、温度、亮度等物理量的传感器，以及煤气浓度等化学量的传感器，其作用如表 1 所示。

Ⅲ 按使用方法分类的传感器

接触式传感器

接触式传感器是固定在被测量的对象上，或借助接触状态来检测的传感器。例如，水电站大坝的混凝土壁上所承受的压力，如图 3 所示，随水位的高度而变化。把坝壁上所承受的压力通过贴在壁上的应变片的变形，变换为电信号进行监视。

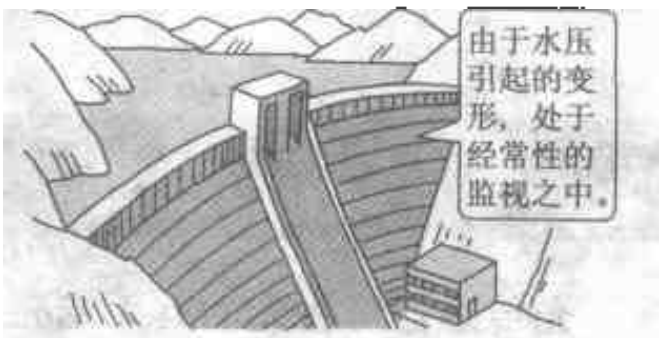


图 3 大坝上承受的压力

接触型传感器的特征为：需承受恶劣的外界环境，可对状态进行直接检测。

非接触式传感器（参见图 4）

如光传感器与温度传感器等，由于与被测量对象没有直接接触，所以有承受环境影响的能力。

要从传感器所获得的信号中取出测量对象的真正数据，是必须下一番功夫的，但由于测量对象不与传感器直接接触，故有不会影响其动作的优点。

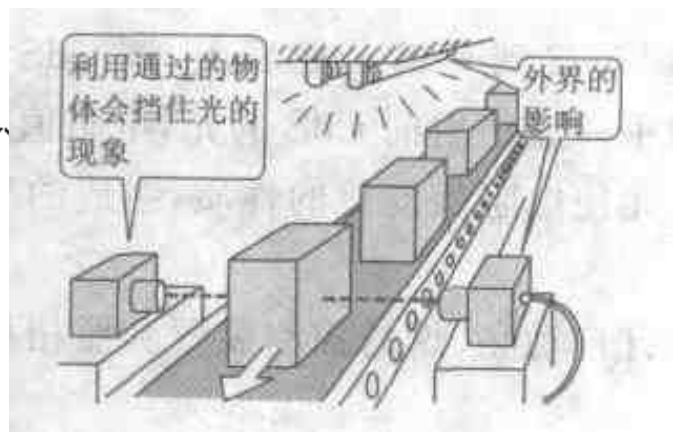
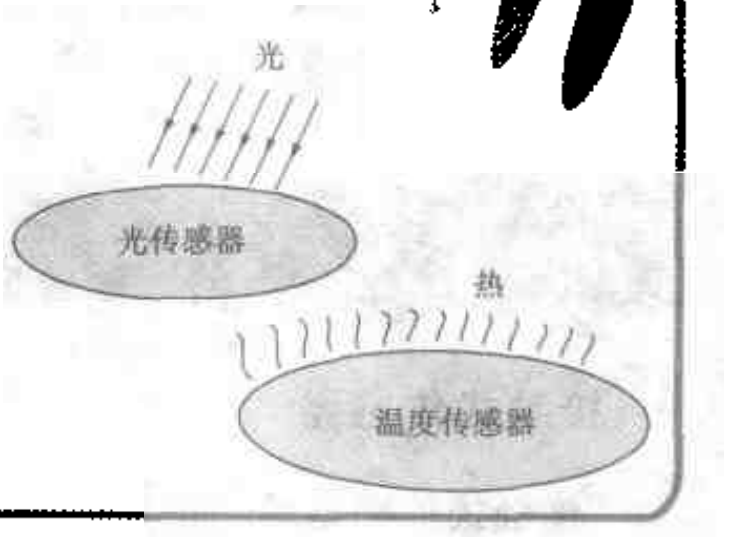


图 4 非接触型传感器（透光型）的例子

2 各种各样的传感器(1)

传感器有很多种，现对其中常用的传感器进行介绍。



I 光传感器

光传感器可将光的有无或亮度等光信息变换为电信号。

硫化镉传感器

CdS(硫化镉)传感器的特征：

① 一般检测动作快；

② 不接触检测对象；

③ 检测时无噪声(噪声是计算机控制中造成误动作的原因)；

④ 检测方便简单；

⑤ 传感器价格低廉。

如图 1 所示，把万用表接到 CdS 传感器上，测 CdS 的电阻时，在明亮的房间里，电阻值非常小，用手一遮住射向 CdS 的光，电阻值就增大了。

图 2 表示利用档住光使电阻值变化的性质，组成的导光型传感器的检测原理。

CdS 元件常用于路灯自动亮、熄的控制装置和照相机自动曝光装置中。

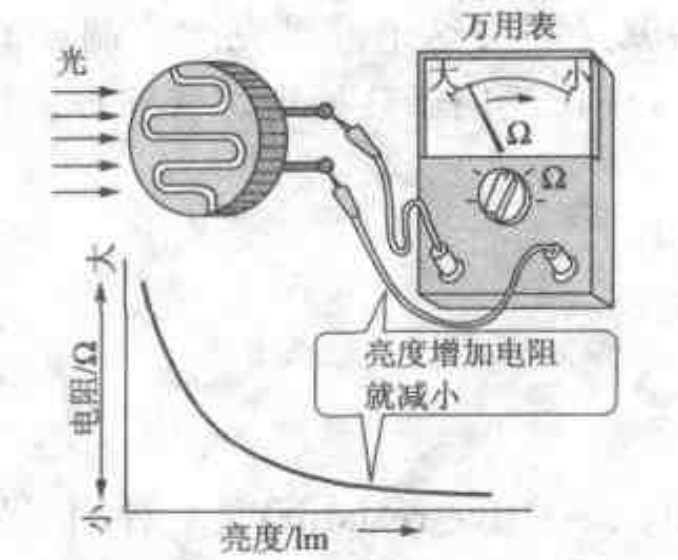


图 1　CdS 等导光型传感器的特征

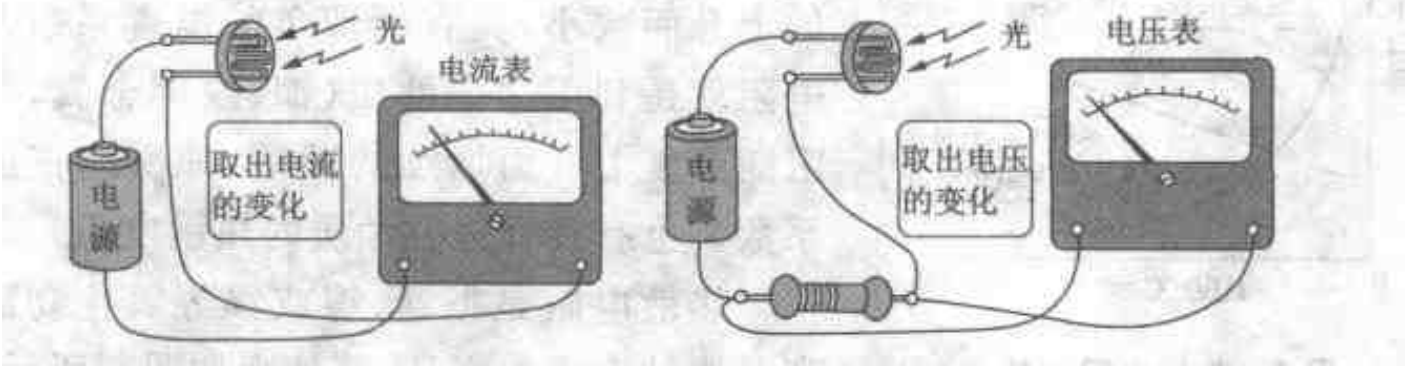


图 2　光传感器的检测原理

Ⅱ　温度传感器

温度传感器在家庭中凡是需要控制温度的设备中，如电冰箱、空调机，电热器具的电暖器、电饭煲等均会用到，是与我们关系最为密切的传感器。

这里介绍常作为温度传感器用的热电偶和双金属温度传感器。

热电偶温度传感器

如图 3 所示，把两种不同金属的两端连接起来，当两端上有温度差时，就产生如图 4 所示的几乎与温差成比例的热电势，这一现象称为塞贝克效应。

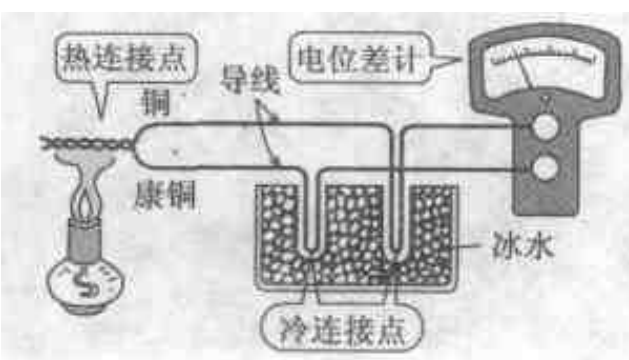


图 3　热电偶温度传感器

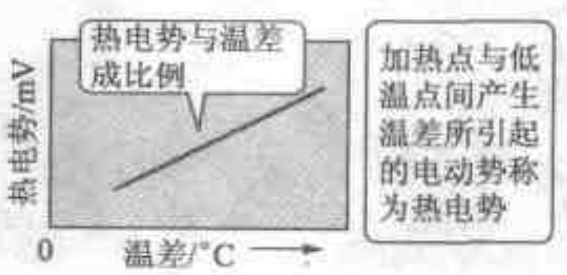


图 4　热电势与温差的关系

热电偶温度传感器就是利用这个塞贝克效应制成的产品。由于热电势数值大而且稳定，所以工业中广泛用于热处理炉管理等。用热电偶可以测量的温度范围如下：

铜-康铜 —200～400℃

铂-铂铑 0～1 600℃

热敏电阻

热敏电阻(thermally sensitive resistor)是一种用半导体的电阻型温度传感器。

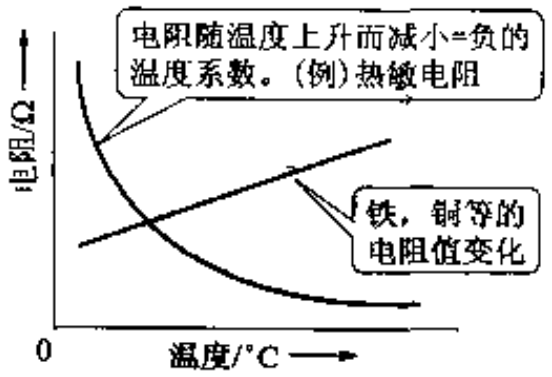


图5 热敏电阻的特性

如图5所示，热敏电阻的电阻值随着温度的上升而减小，具有所谓负的温度系数。热敏电阻就是利用这一性质(铜、铁等金属，具有电阻随温度上升而增加的性质，即所谓正的温度系数。也有具有这种性质的热敏电阻)。

热敏电阻是把锰、镍或镉等氧化物混合经高温烧结而成的产品，其外观如图6所示。如图7所示，在温度变化时，热敏电阻可用万用表测量其随之而变化的电阻值。

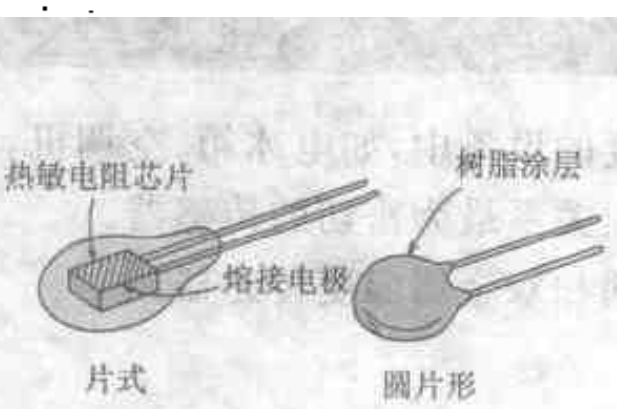


图6 热敏电阻的形状

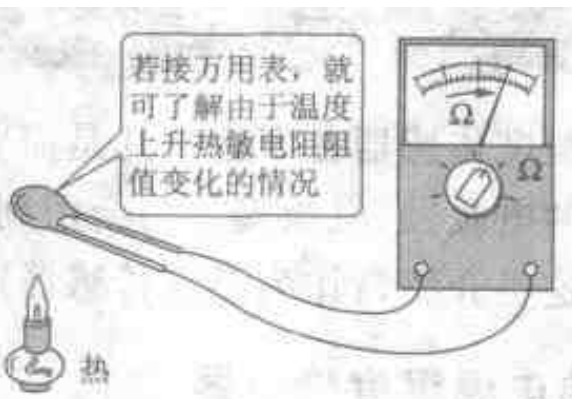
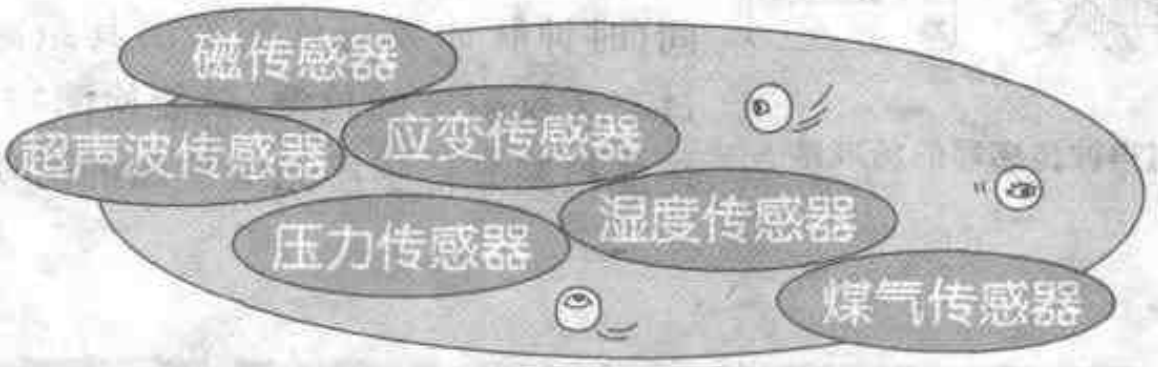


图7

3 各种各样的传感器(2)



Ⅰ 磁传感器

磁传感器是检测磁性的传感器。这种传感器有以下两类：

① 利用磁铁吸铁时的磁吸力作用。例如簧片开关。

② 将由于电流流过电线或线圈形成的磁场变换为电气量的器件。例如霍尔元件。

因为磁传感器能以非接触形式检测，可用于无触点开关、车用发动机等的车速传感器、物体位置检测器等。

图 1 是铁路常用的数字式轨道控制簧片开关的例子。线路上设置簧片开关，当装有磁铁的列车在上面通过时，以簧片开关的开、闭来检测列车通过与否。

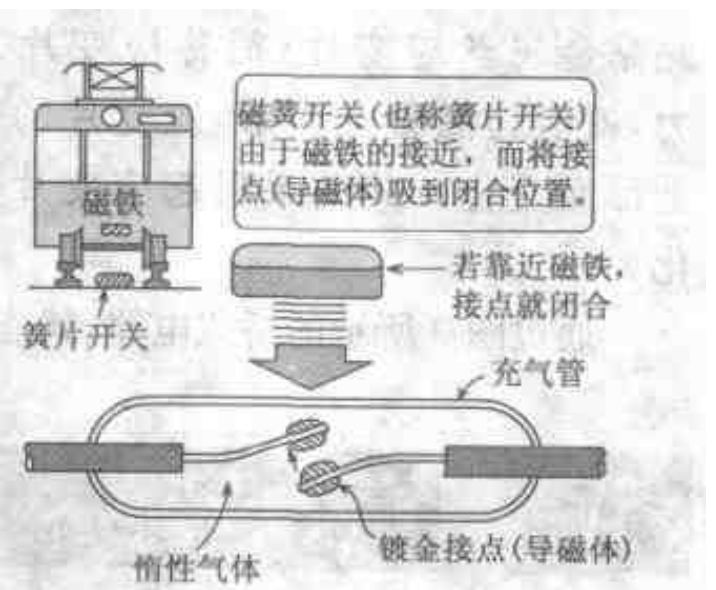


图 1　利用簧片开关

Ⅱ　超声波传感器

超声波传感器是测量离反射体之间的距离的传感器。

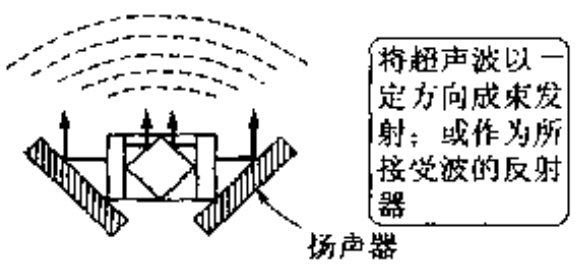


图 2　超声波传感器的扬声器

如图 2 所示，发出超声波的扬声器，把超声波发送器与接收器组合在一起配置。测量从发送器发出超声波到接收反射波为止的时间(时钟脉冲数)，就能算出其距离。并且通过有无反射波的检测，就能检测物体的位置，还能检测光传感器无法检测的透明瓶子的有无。

Ⅲ　应变传感器

为检测由外力引起混凝土等的应变，可用应变传感器。混凝土是由称为骨料的砂与卵石、水泥，加水等拌合制成的，主要用于承受压力的场合。其骨料的比例不同，变形程度有变化，为此必须知道骨料的比例，混凝土能产生什么样的变形(应变)。

现在，在图 3 所示的混凝土压缩试块上黏贴金属丝应变片(简称应变片)，一加上力，就引起混凝土变形，应变片的长度随着变形而变化。其结果引起应变片阻值的变化。

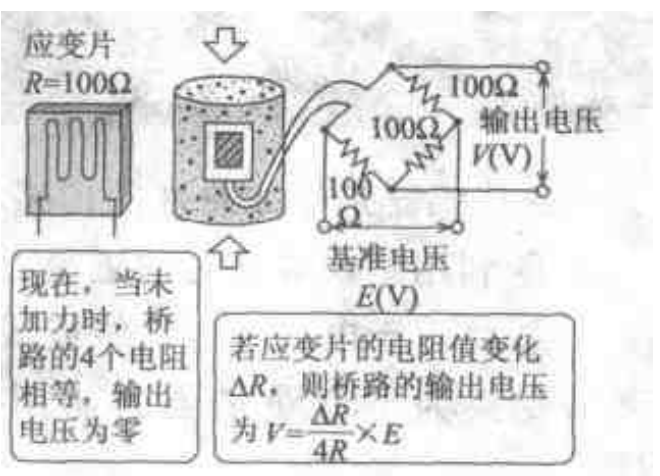


图 3　应变传感器

通过图 3 所示的桥式电路，就能测出由其阻值变化而引起的输出电压的变化。

Ⅳ　压力传感器

图 4 所示的半导体压力传感器，是利用在硅半导体晶体上加上压力造成变形而产生电荷的压阻效应。也就是说，把压力变成电信号。这种传感器的尺寸小、灵敏度高，且能稳定地检测出反复的变形。并且，这种传感器还能用于检测电气吸尘器的吸入压

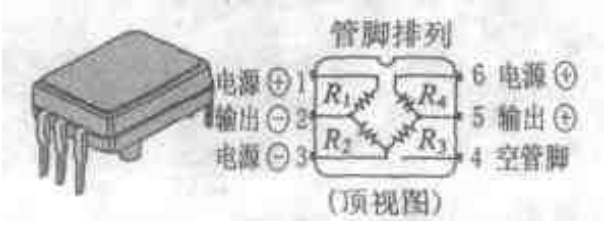


图 4　半导体压力传感器

力。

Ⅴ　湿度传感器

湿度传感器是检测大气中所含水蒸气比例的传感器。此传感器的工作原理是把附在陶瓷片上的水分变化变换为电信号，该陶瓷片就是湿度传感器的检测元件，它常用于空调器，农用大棚等场合的湿度控制，图 5 表示陶瓷型的湿度传感器。

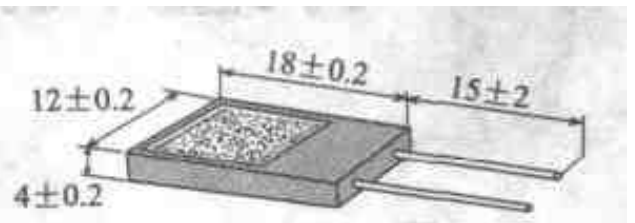


图 5　湿度传感器

Ⅵ　煤气传感器

煤气传感器是检测大气等气体中所含煤气成分浓度的传感器。

家中所用的煤气泄漏报警器用的也是煤气传感器。若用电吹风把煤气吹入煤气泄漏报警器，报警器就会发出“蜂鸣声”，报警煤气泄漏。

图 6 表示出于煤气分子吹到了煤气传感器的半导体上，从而引起了电阻变化，这就是煤气传感器的测量原理。



图 6　煤气传感器的测量原理

4 来自传感器的输入信号



已介绍过各种传感器的种类与检测原理、特性、用途等。

这里探讨一下来自传感器的检测数据是以怎样的形式输入计算机的。



Ⅰ 数字量输入

例如，如图 1 所示，来自按下按钮开关的信号，可直接得到“ON(通)＝1”，“OFF(断)＝0”的数字量信号。得到这种数字量信号的传感器，有检测机床工作台位置用的限位开关与磁性传感器的簧片开关等。

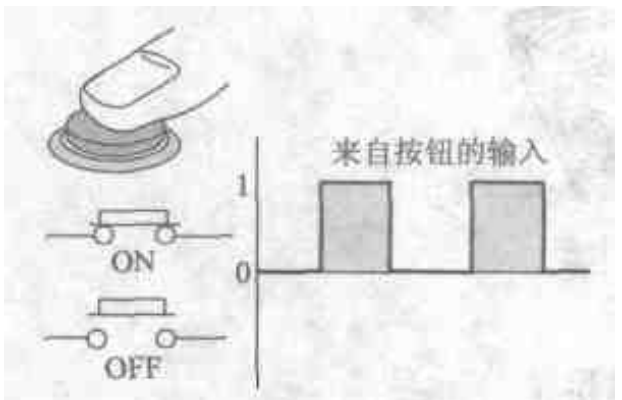


图 1　按钮开关

这样的数字量，其本身就可以用计算机处理，但实际上，来自外部的噪声之类会混入输入数据中，多数情况下，达不到计算机处理中所必要的电信号电平，因此要对此采取必要的措施。

Ⅰ　模拟量的输入(A/D转换)

现在来考察--下路灯的自动控制开关，这种自控开关，根据光传感器检测到的亮度，输出通、断值的信号，就可达到控制目的。

为此，如图2所示，只要通过检测光强弱的水平来操作亮灯、熄灯的较简单方法，即可把信号传给计算机。

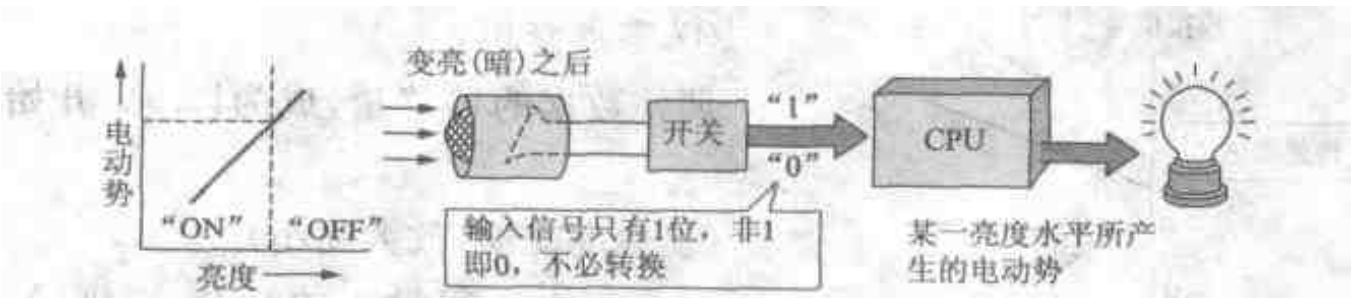


图2　模拟量的二值输入

但是通常亮度、温度之类的量是连续变化的数据(这样的数据称为模拟量)。若要把对应其亮度或温度水平的数据，作为计算机能处理的量，则必须如图3所示，把对应于模拟量大小(强弱等)的量变换为计算机能处理的数字量。

这种把模拟量变换为数字量的过程称为A/D(模/数)转换。通常，来自传感器的输入电信号，要通过图3所示的放大器，放大为0～5V的范围，再输入A/D(模/数)转换器。4位A/D转换器可输出16种，8位A/D转换器可输出256种二进制数字量信号数据。

也就是说，要使传感器检测的亮度或温度等信号转换为二进制数字信号来输入计算机。

图3所示为与亮度相对应的A/D

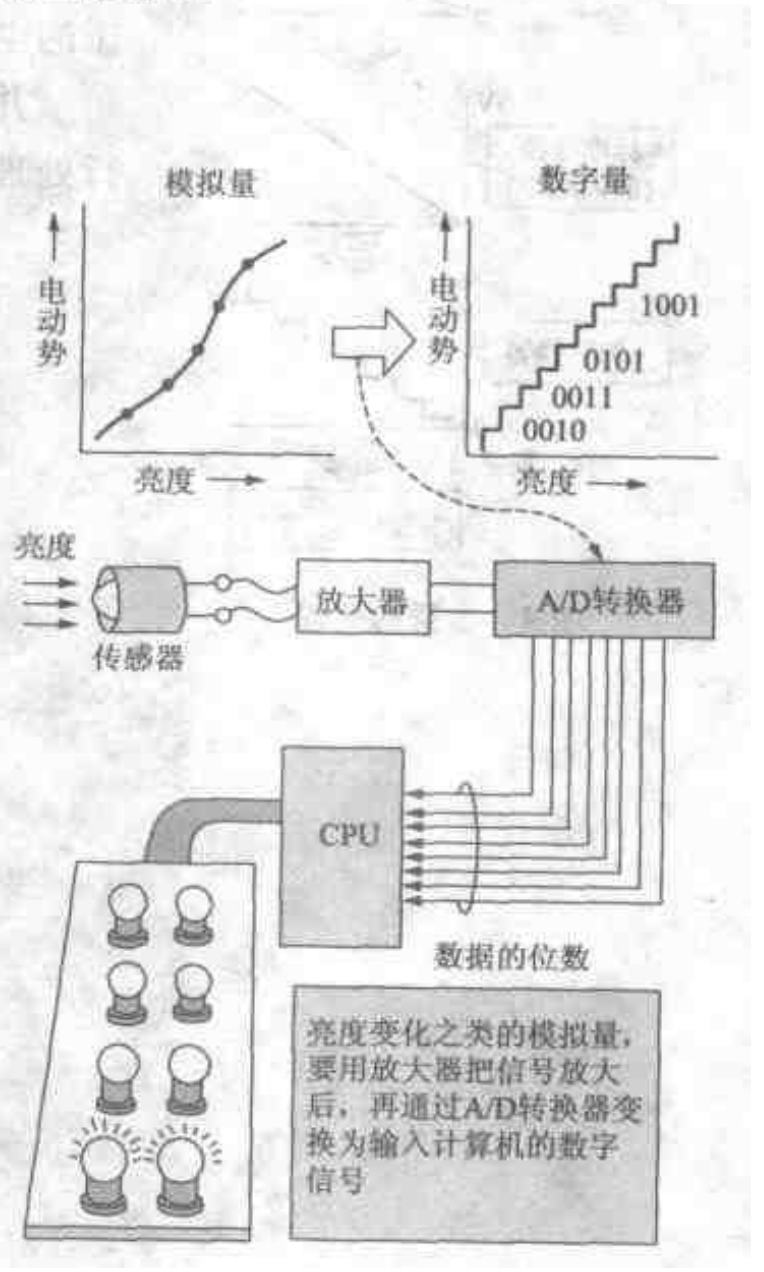


图3　A/D转换

信号转换的例子，由 8 位数字信号显示两个灯亮熄的状态。

Ⅲ　数据的处理

A/D 转换是一种如图 4 所示的数据处理过程，

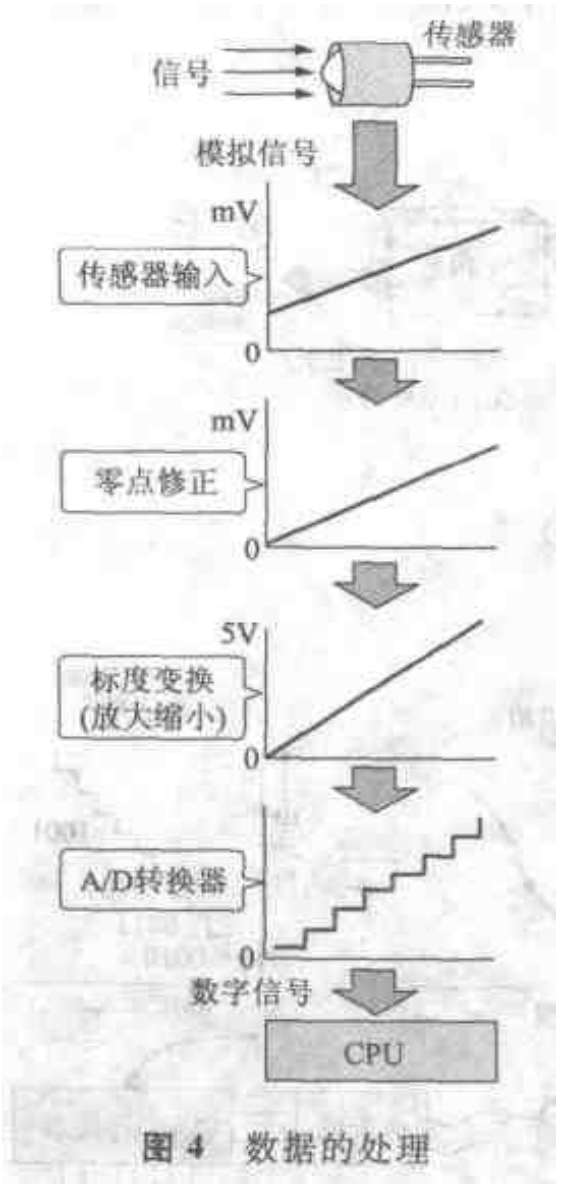


图 4　数据的处理

当需要把输入到转换器中电压的大小调整到某一范围时必须进行这样的转换。

① 零点修正

调整数据的偏移量，成为以 0V 开始的信号。

② 标度变换(放大缩小)

在模拟电压很微小的场合，要把 A/D 转换器的输入电压放大调整为 0～5V 大小可分压的电压。

并且，检测数据中有噪声窜入时，必须进行处理，把它滤去。

5 数字量的输入

I 从传感器向计算机的输入

前面已介绍过了关于位移传感器、磁性传感器、温度传感器、光传感器等各种各样的传感器。这里探讨一下，如图 1 所示，这些传感器所得到的信息（检测的结果），如何与计算机连接（输入）。也就是说，在计算机控制中使用传感器的方法。这里所介绍的只不过是其中一个例子，大家在自己实现控制的过程中，一定要用心调整传感器的输入。

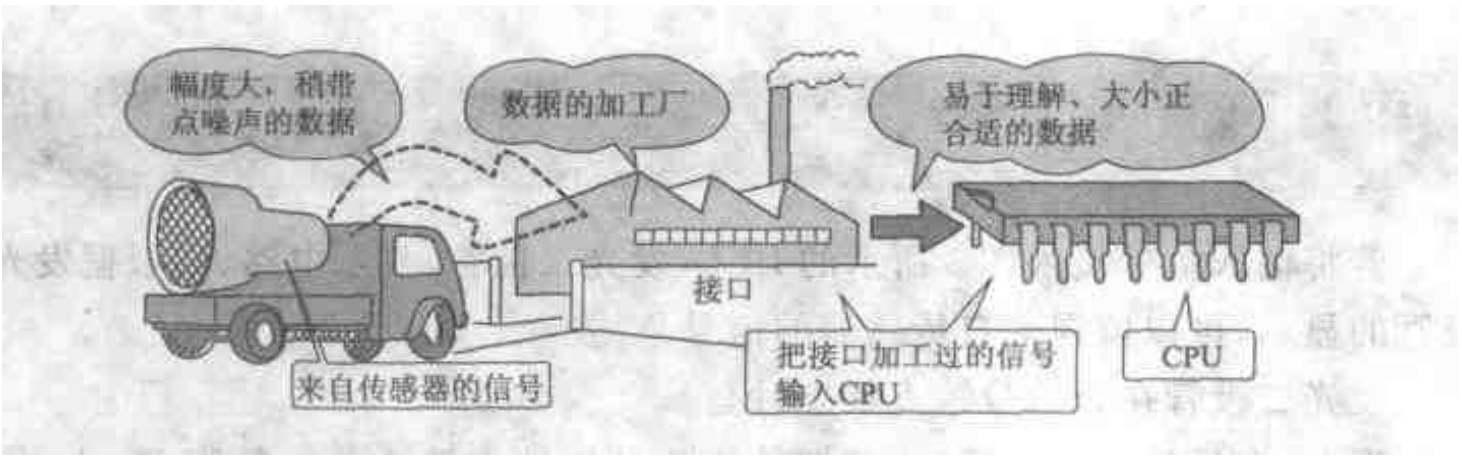


图 1 来自传感器的输入信号的传输

从传感器向计算机的输入，必须用图 1 所示的接口（能很方便地实现计算机信号与输出设备信号的交换所用的装置）。此接口中，有信号输入、输出口，其输入口就是用于输入来自传感器信息的。

Ⅱ　开关的输入数据

按下按钮开关或乒乓开关，输出的通、断状态就是开关的输入数据，即开关量。

控制的输入，最多的就是按下按钮或通过手柄位置切换等所检测到的开关通、断状态。从这一意义上来讲，开关也可以包括在传感器的类别中。

图 2 表示乒乓开关的输入电路。图中，在开关接通时，计算机输入为[0(V)="0"]，开关断开时，计算机中输入为[5(V)="1"]。

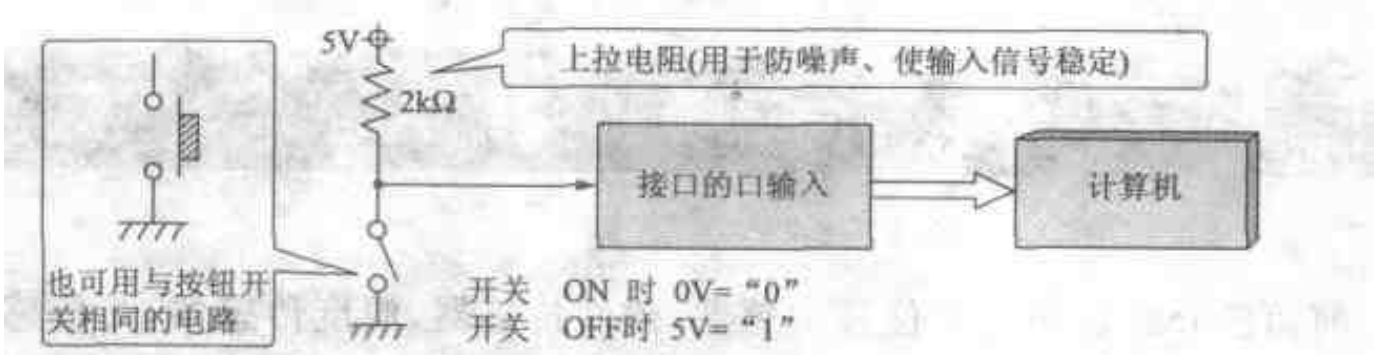


图 2　乒乓开关的输入电路

电路中连接的电阻称为上拉电阻，可防止无输入时噪声等引起的错误输入。

Ⅲ　输入信号的监视

若将输入信号接入图 3 所示的 LED(发光二极管)显示电路，则根据发光二极管的显示，可以监视来自传感器的信号。

发光二极管在无信号输入状态时亮。

现在，在乒乓开关接通时，若把输入钩针连接在第 2 位的管脚 D_1 上，则由电路可知，第 2 位的输入信号为"0"，因此，第 2 位的发光二极管熄灭，就可确认其输入为"0"。

通过这样连接到任意位上，发光二极管的显示状态就产生变化，因此开关(传感器)的状态，就可以通过发光二极管的显示而得到确认。

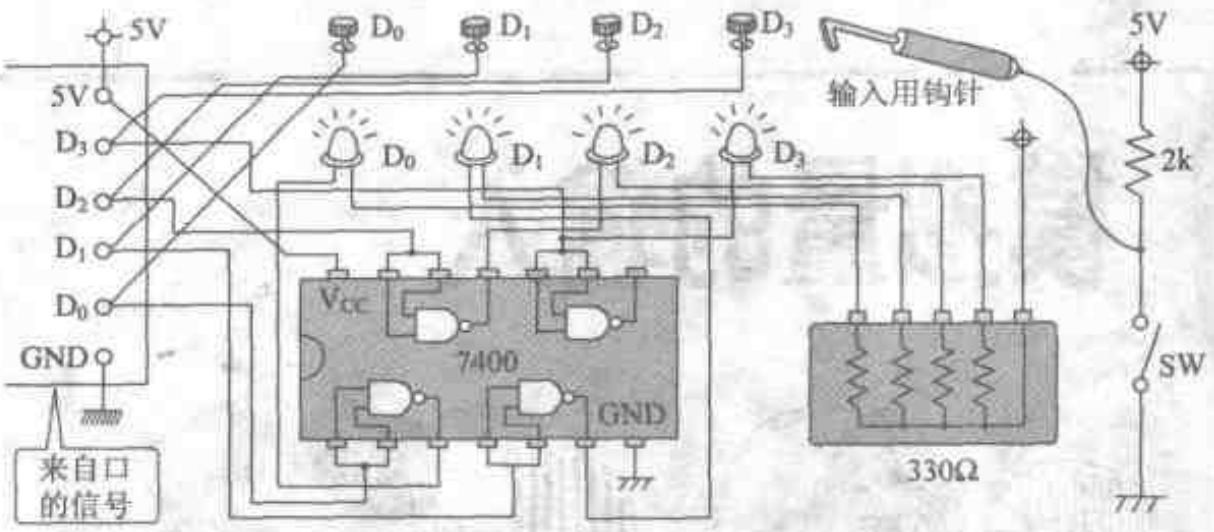


图 3　用发光二极管的输入信号监视

5.1.2 输入电路

现在把开关输入电路分成四路，接入发光二极管的监视电路中，把对应于某一时刻开关的通断状态的信号，接入插口中，并输入到计算机中。

例如，第 0 位和第 2 位开关接通时，发光二极管的状态如表 1 所示。因此，如图 3 所示的 4 位电路，以各种不同的开关(传感器)信息，可得到表 2 所示的 16 种组合，其值可作为来自传感器的检测数据输入计算机。

可把这些数据作为目标控制的基础。

表 1　4 位的输入数据

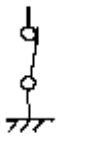
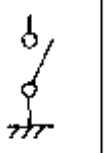
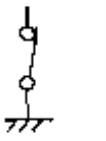
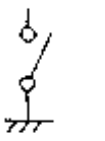
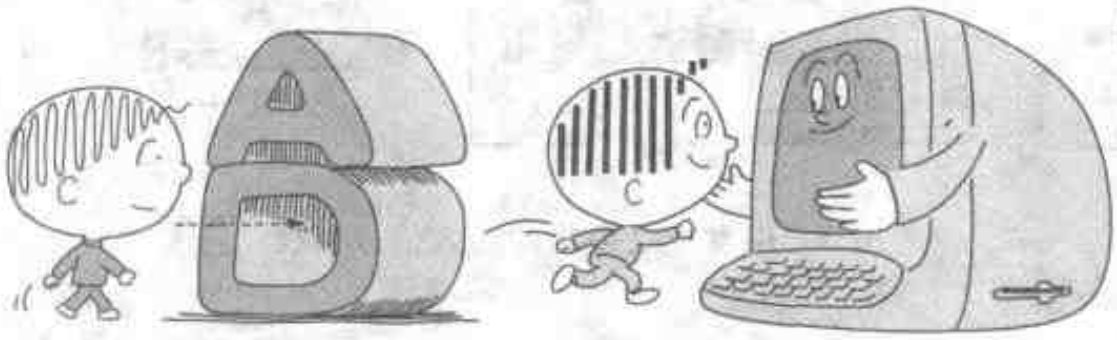
位号	第0位	第1位	第2位	第3位
开关状态				
信号	"0"	"1"	"0"	"1"
发光二极管的显示				

表 2　4 位数据

位号	0	1	2	3
LED ○ 灯 ● 亮	○	○	○	○
	○	○	○	●
	○	○	●	○
	○	○	●	●
	○	●	○	○
	○	●	○	●
	○	●	●	○
	○	●	●	●
	●	○	○	○
	●	○	○	●
	●	○	●	○
	●	○	●	●
	●	●	○	○
	●	●	○	●
	●	●	●	○
	●	●	●	●

6 模拟量的输入



Ⅰ 从传感器取得模拟量

要将模拟量输入计算机，必须将模拟量转换为数字量即实际上，来自传感器的信息不是直接输入到计算机的，如图 1 所示，要通过 A/D 转换器进行上述的模/数转换。

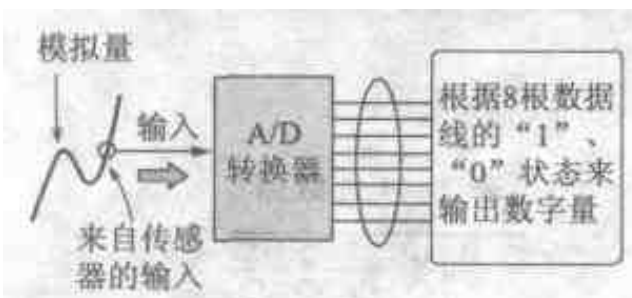


图 1　A/D 转换器的组成

也就是说，只要考虑模拟量怎样从传感器进入 A/D 转换器就可以了。

Ⅱ A/D 转换器(ADC0809)的定义

以 A/D 转换器中常用的 ADC0809 为例，其组成如图 2 所示。

图 2 是把热敏电阻的模拟量作为从 IN0 等管脚的输入，把基准电压上限 5V、下限 0V 之间分为 256 等分作为数字量数据，输入到计算机的输入电路的例子。施加在 IN0～IN7 这几个管脚上的电压变化状态，就相当于来自传感器的模拟量。

IN0~IN7	可进行8通道的模拟量输入
ADD A ADD B ADD C	上述的IN0~IN7的选择，给出所选通道的二进制数
START	若给出“0”~“1”~“0”的信号，就开始转换
EOC	以0~5V范围的数据，作为计算机的输入数据
CLOCK	提供时钟脉冲
REF(+)	基准电压
REF(–)	给出来自传感器的输入电压的上限与下限
DATA0~7	数字量信号的输出

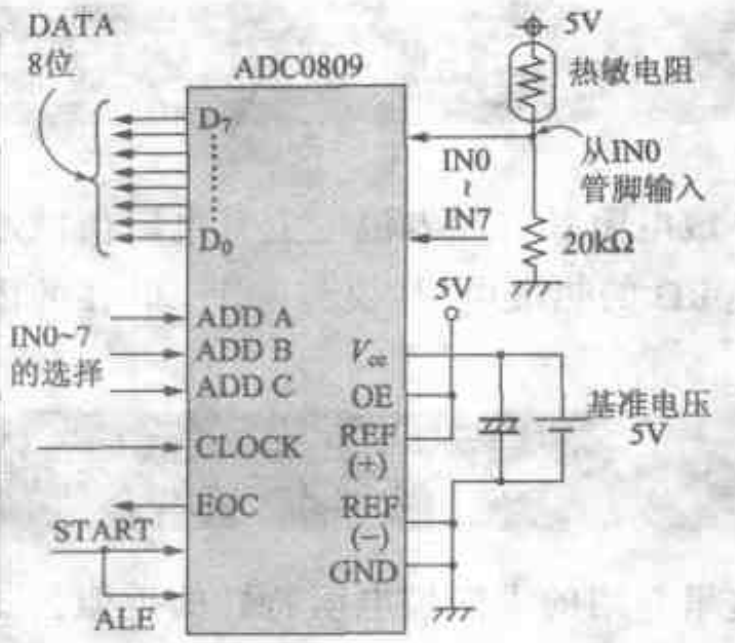


图 2 A/D转换器(ADC0809)

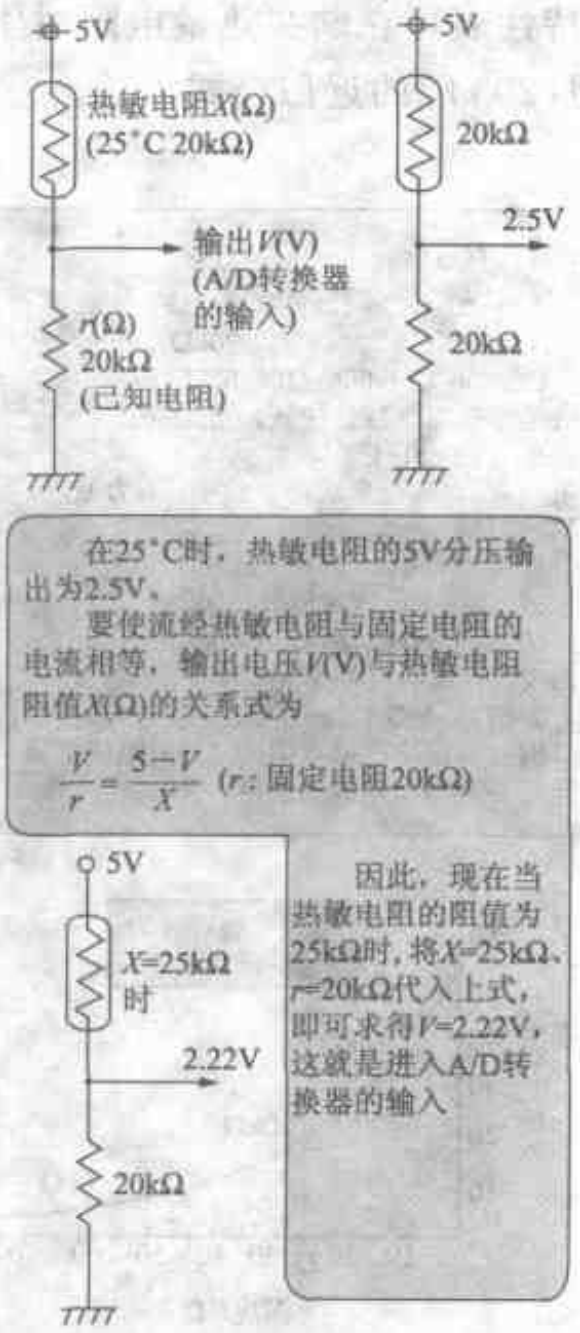


图 3 用热敏电阻检测温度的电路连接

Ⅲ　用热敏电阻检测温度的电路连接

热敏电阻具有随着温度上升电阻值减小的性质，因此如图 3 所示，在 25℃时为 20kΩ 的热敏电阻，要与一个 20kΩ 的固定电阻相串联。

Ⅳ　求传感器特性的校正近似公式

这里介绍的求热敏电阻特性的近似公式，对其他的传感器，也同样适用。

如图 4 所示，热敏电阻阻值与温度的关系肯定不是线性的，因此，要用热敏电阻的阻值与温度的特性表，画出阻值温度特性曲线，求得与此曲线近似的公式，再求出实际的温度。特性表可在购买热敏电阻时索取。

图 4 给出了热敏电阻(203T)的近似公式求法。

由特性表可得：

A点 10°C 36.12kΩ取 36kΩ
B点 25°C 20kΩ
C点 60°C 6.006kΩ取 6kΩ

通过这三点的特性曲线，可用在式

$H=a/(X+b)+C$

中代入a、b、c来求得。式中，H为温度；X为阻值。

10°C时 $10=a/(36+b)+c$ ①
25°C时 $25=a/(20+b)+c$ ②
60°C时 $60=a/(6+b)+c$ ③

由式①，②，③可解出：

$a=613$，$b=3.4$，$c=-5.6$

因此，根据热敏电阻(203T)的阻值，就能求得该阻值相应的温度

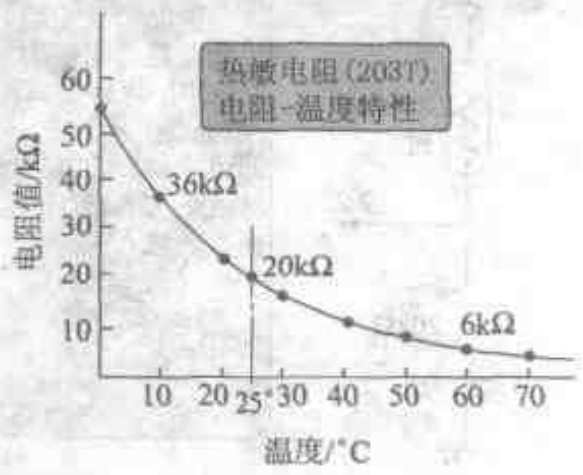
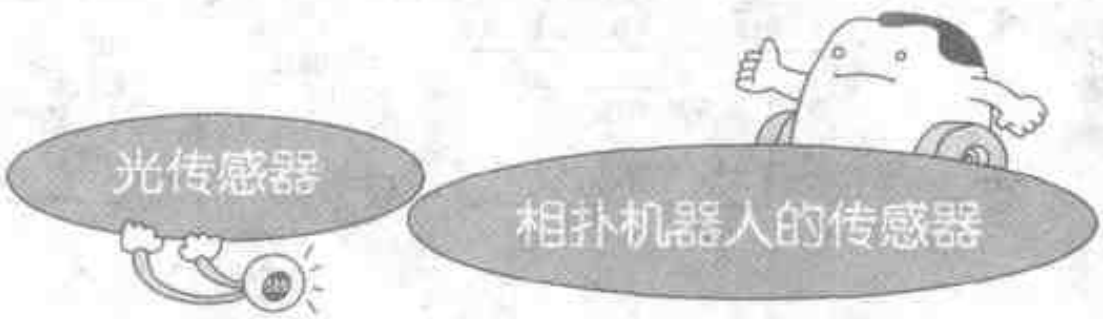


图 4　近似公式的求法

7 传感器输入实践



1 转换模拟量的开关

举例介绍通过光传感器来对模拟量的某一变化值进行通、断操作的用法。

硫化镉传感器的阻值随光通量(亮度)而变化。在如图 1 所示的电路中,将这一变化与变阻器(V_R)组合,加到晶体管上作为控制电压,就可以将晶体管作为开关使用。

把这一电压作为控制信号的开关量,输入计算机的话,就能通过亮度的变化实现开关输入。其开关状态,可根据发光二极管 LED 的亮、熄得到确认。

现在来研究一下图 1(a)电路中,关于硫化镉 CdS 传感器的受光状态引起的动作。假定变阻器的阻值为 50kΩ,根据下式可求出 A 点的电压:

$$V_A = V_{cc} \times R_x / (R + R_x)$$

则 A 点的电压:

亮时　　0.28V

暗时　　4.6V

因此,暗时(光被遮住)加在基极上的电压增大,晶体管导通,发光二极管 LED 亮,同时从输出管脚 B 输出"1"(当 B 点电压为 0V 时,定义为"1")。

图 1(b)为受光时硫化镉 CdS 传感器起作用而输出"1"的电路。此电路中,受光时 CdS 阻值减小,基极电压增大,点 B 的电压为 0V,即输出为"1"。

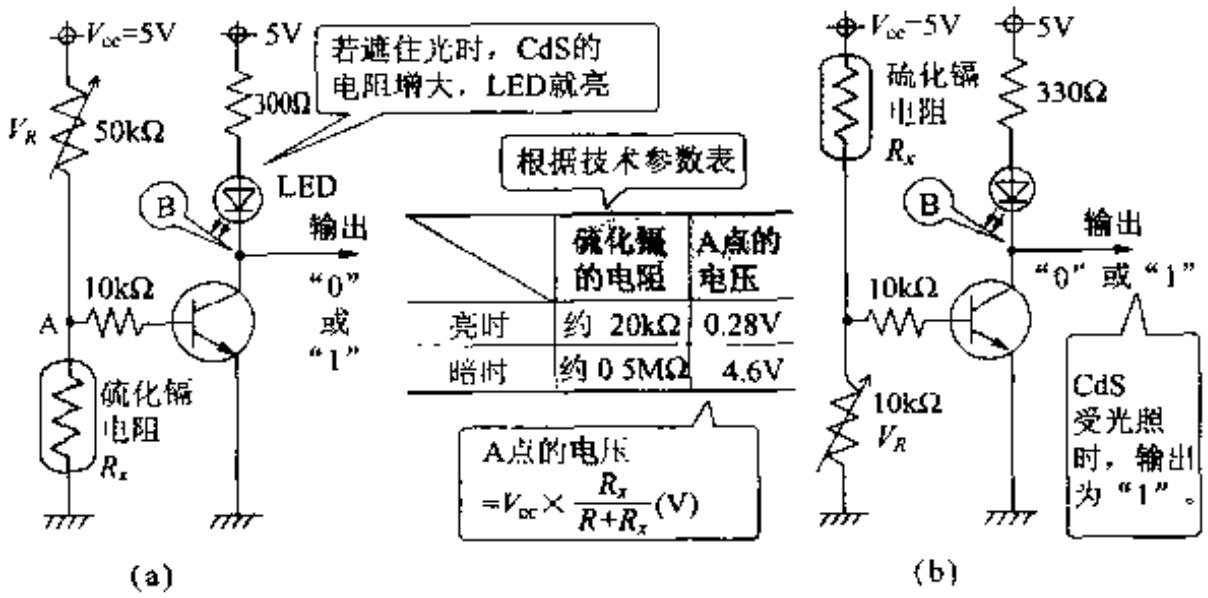


图 1　根据光通量组成通、断电路的例子

2　相扑机器人的传感器

"制作相扑机器人"是最近学习机电一体化非常适合的入门途径，越来越引起大家的重视(参见第 9 章)。机器人相扑的要点之一就是在检测到赛场的边线白线之后，要让驱动电动机反转，使机器人退回。我们来介绍一下白线的检测传感器。

赛场边线(白线)检测传感器

相扑机器人的总体状态与大小均受到限制，因此对其结构零件也力求紧凑。这里选用的是欧姆龙公司的反光型传感器(型号 EE-SF5B)。如图 2 所示，这种传感器因尺寸小，可用黏接剂或简单的安装件装在相扑机器人上。

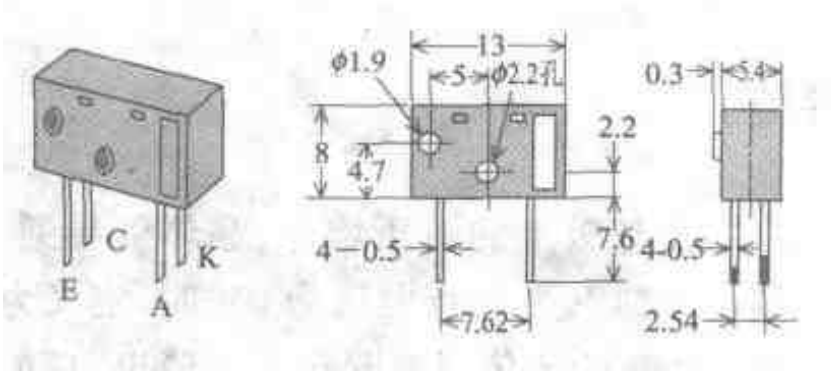


图 2　EE-SF5B 外形尺寸与管脚排列

动作原理

如图 3 所示，用红外发光二极管作为光源，赛场白线的反射光照到光导晶体管上，就使开关动作。根据制造厂的样本可知：

检测距离　5mm 内可检测反射率
为 90%的白色类物体

这种反光型传感器是大小与性能最合适的传感器之一。

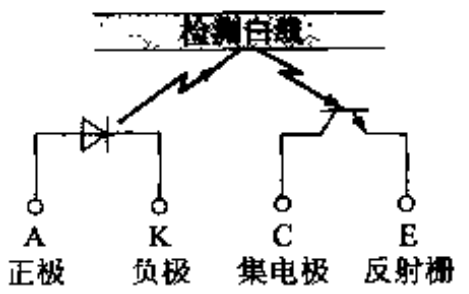


图 3　白线的检测

图 4 所示为相扑机器人的赛场白线检测电路的例子。用 30kΩ 的可变电阻来调整对赛场白线的敏感度。来自这些传感器的输入信号，通过 8255 接口输入计算机。由这一信号可控制左右电动机的正、反转。输入电路中串入两个反相器，目的是为了稳定来自传感器的输入信号。

传感器电路中，如此例所示的，通过电路处理，使输入信号稳定，并加强抗干扰能力，也是很重要的技术之一。

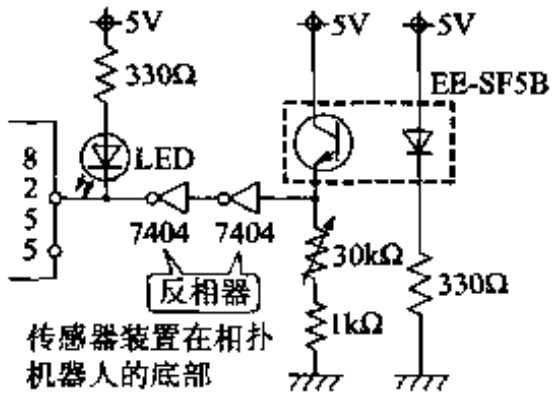
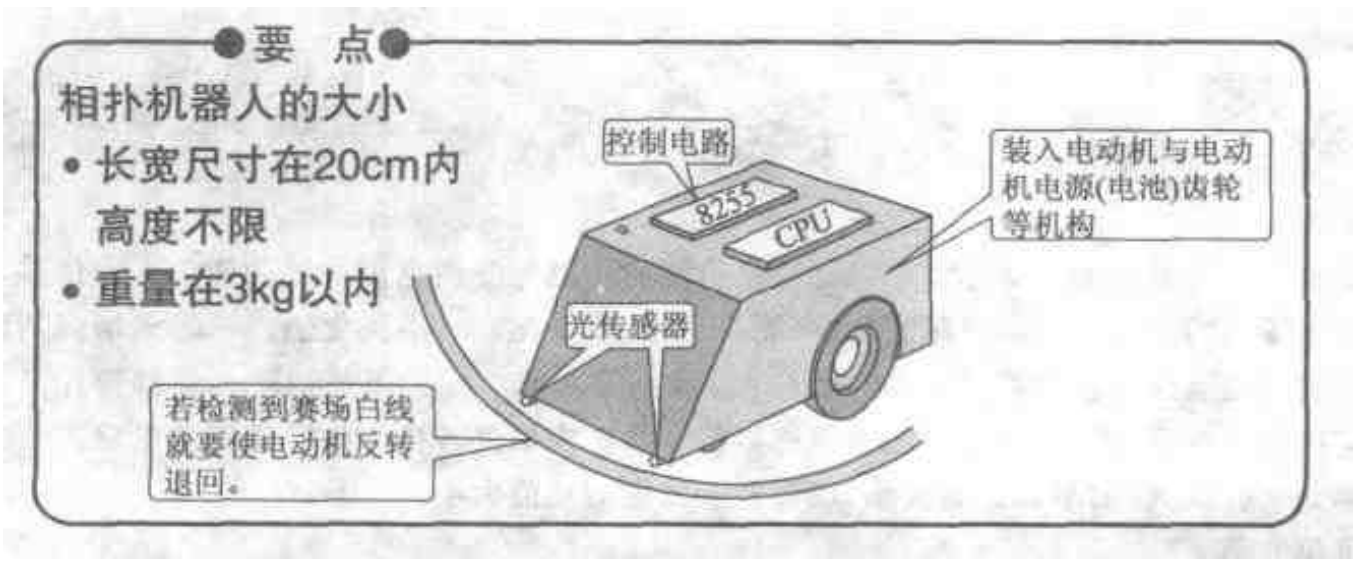


图 4　赛场白线检测电路



Q 考试大纲

① 传感器有什么类似名词?
② 列举相当于人的器官的传感器。
③ 传感器的两大功能是什么?
④ 列举传感器检测的物理量。
⑤ 列举传感器检测的化学量。
⑥ 列举把传感器信号变化为电信号功能的例子。
⑦ CdS 是什么东西?
⑧ CdS 受到光照后电阻值会有什么变化?
⑨ 列举两种温度传感器。
⑩ 热电偶以什么原理检测温度?
⑪ 当温度上升时,热敏电阻的阻值会有什么变化?
⑫ 受磁场影响时,接点会通断的开关称为什么?
⑬ 应变片是用来测量什么的?
⑭ 利用压阻效应的传感器叫什么?
⑮ A/D 转换电路是用来干什么的?
⑯ A/D 转换过程中,零点修正是何意?
⑰ A/D 转换过程中,标度变换是何意?
⑱ 8 位 A/D 转换器可转换多少种相应的数据?
⑲ 叙述一下热敏电阻的温度检测电路组成。
⑳ 传感器的修正近似公式有何必要?

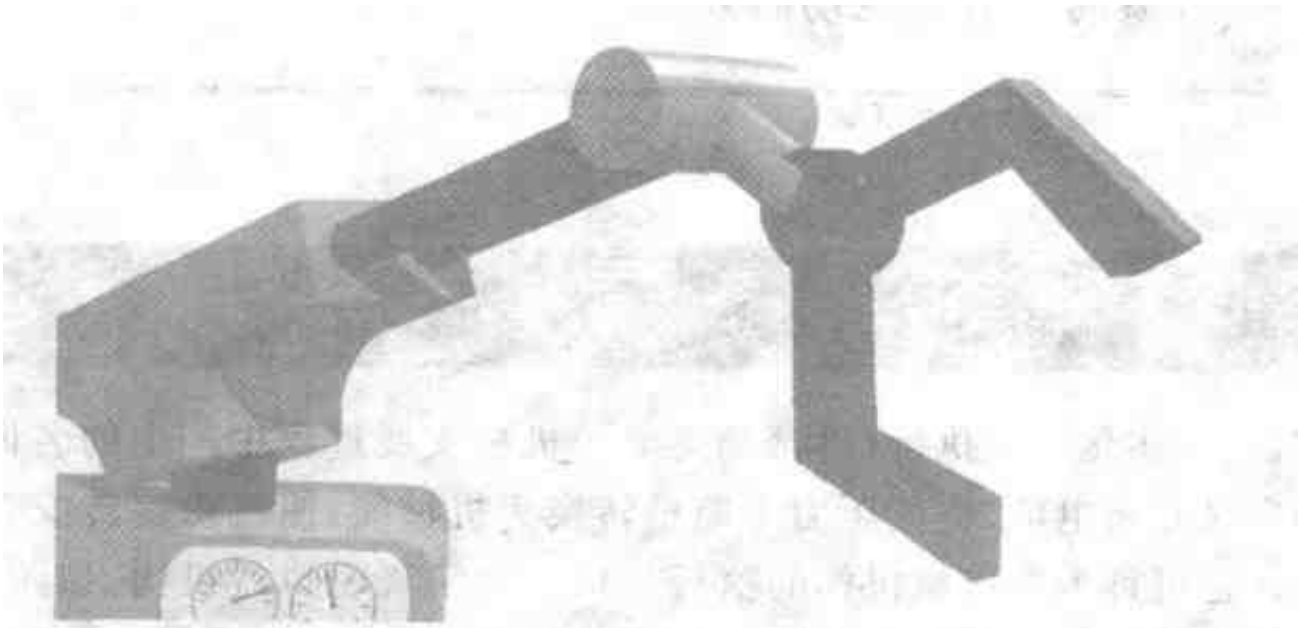


A 答　案

①转换器。　②参见第 2 章扉页。　③信息检测的功能与变换电信号的功能。　④位移、速度、力。　⑤煤气的浓度。　⑥电压变化、电流变化、频率变化。　⑦硫化镉。　⑧光增强则阻值减小。　⑨热电偶温度传感器、热敏电阻。　⑩参见第 2 节。　⑪阻值减小。　⑫簧片开关。　⑬力等物体的变形(应变)。　⑭压力传感器。　⑮把模拟量转换为数字量用的电路。　⑯使数据偏移。　⑰放大、缩小数据的大小,成为易于处理的幅值大小。　⑱256 种。　⑲参见第 6 节。　⑳参见第 6 节。

第3章

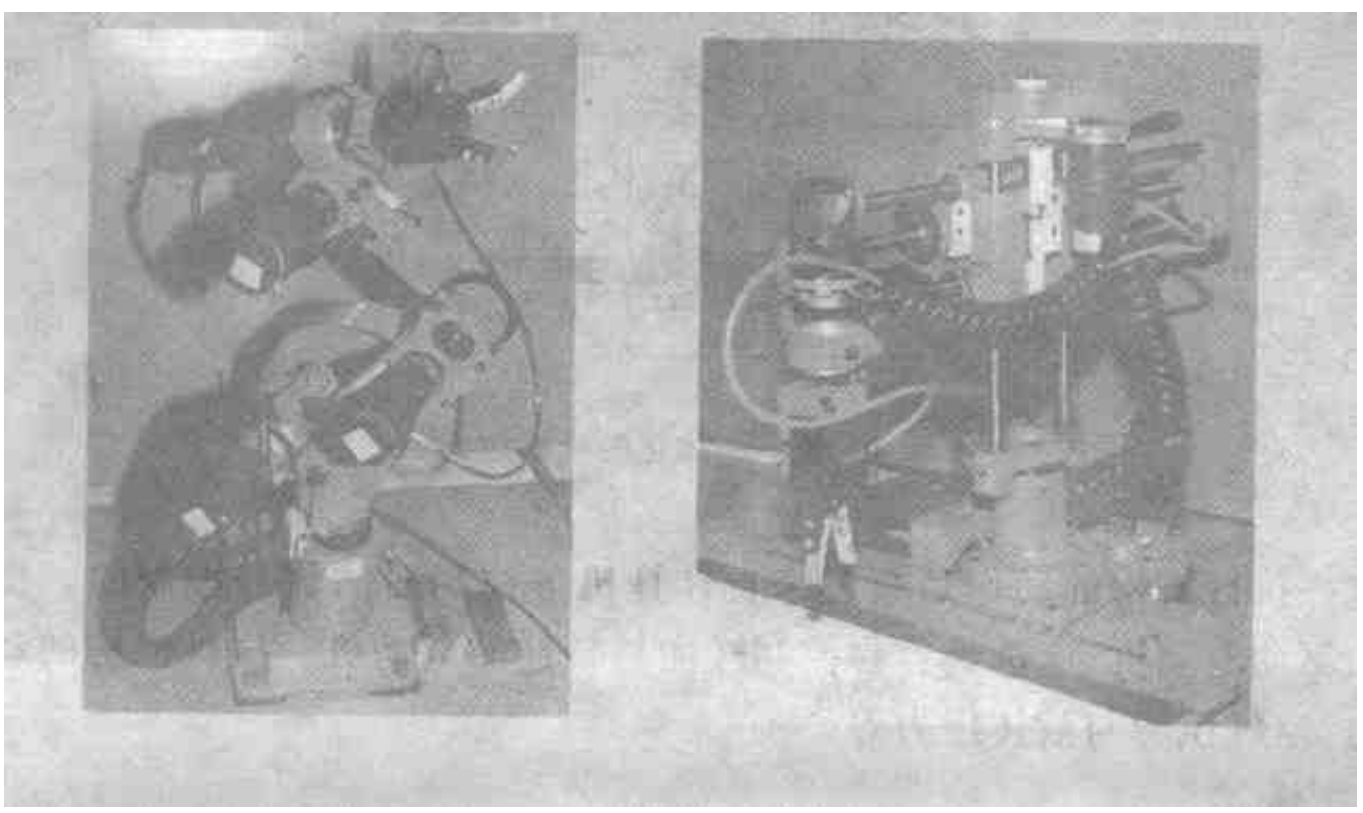
执行机构的基础知识



执行机构的组合与机器人

在机电一体化中将执行机构组合起来就是机器人。经观察分析可知机器人的基本构造部件是电动机械和油缸等执行机构组成的，是由一个一个动作部件构成的。

换言之，机器人也可以称为执行机构的组合体。这里，可以通过观察机器人的动态照片来加深对执行机构的理解。



1 执行机构的作用

执行机构是将能量转换为机械能，使机构作旋转、直线运动的装置。

1 执行机构的定义

“机电一体化”或“执行机构”都是电子机械发展过程中产生的名词。执行机构的定义是将电能、空气压力等能量转换为机械能，使机构旋转或直线运动的装置。也可称为含机械机构的执行机构。为了说明执行机构，如图 1 所示，现在来考察一下应用机电一体化的机械的构成。

图 1 所示为传送皮带上的物体移动的状态，可通过传感器来识别，然后用计算机对此状态进行控制。驱动这一机械系统（传送皮带）的电动机，就是执行机构。

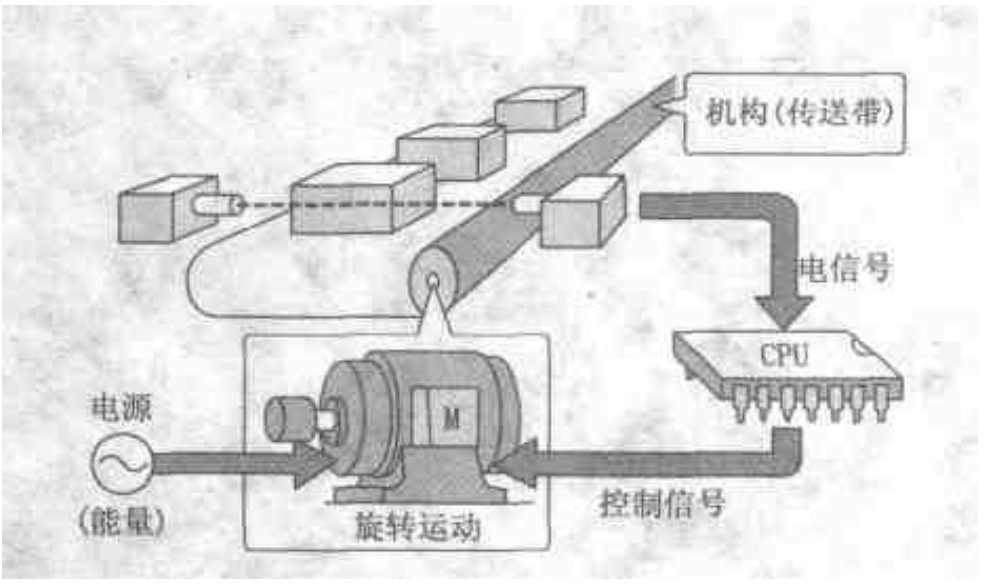


图 1　机电一体化的机械构成

这一执行机构的动作是通过来自计算机等控制器的信号控制的。机电一体化的驱动电动机，应选择容易控制转向与转速的电动机。执行机构的动力源是电力、空气压力与液体压力等。

Ⅱ 执行机构的运动

执行机构的动作可粗略地分为旋转运动与直线运动两类。

旋转运动

通过执行机构，就得到了轴的旋转运动。

直线运动

执行机构的动作是像汽缸那样的直线往复运动。

① 汽缸

汽缸是将压缩空气的能量转换为直线运动的装置。如图 3 所示，它由缸体，活塞及活塞杆等组成。其原理是靠电磁阀切换空气的流向，使汽缸往复运动。

② 油缸

油缸基本上与汽缸相同，是将油压的能量转换为直线运动的执行机构。用电磁阀来切换油的流向，从而产生往复运动。

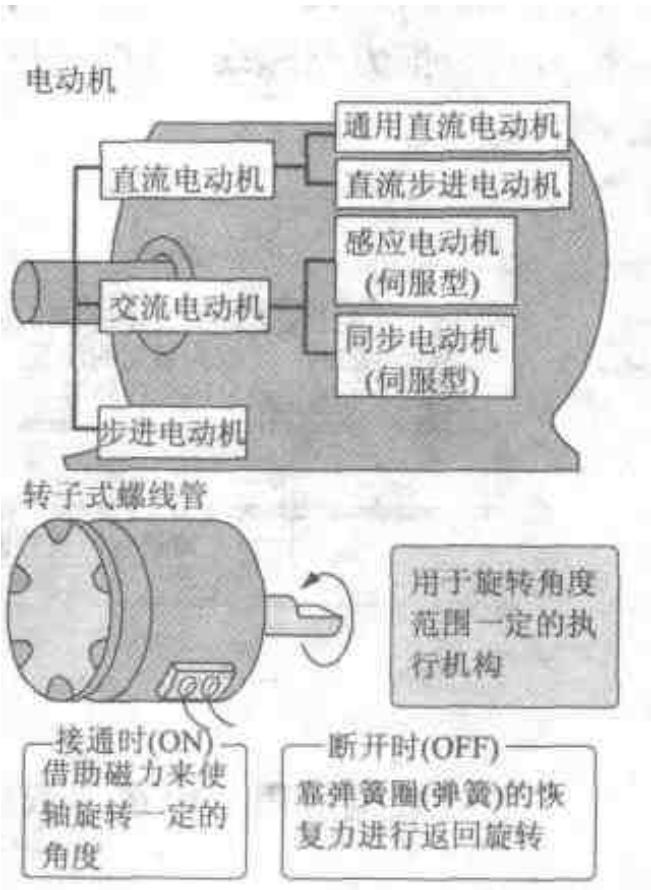


图 2 转子式螺线管的驱动

传递压缩力的媒介是液态油。液体(油)与气体(空气)不同，如进行像图 4 所示的压缩，其体积也不会变化(这一性质称为“液体的不可压缩性”)。因此，可将活塞准确地停留在任意位置。这是油缸的最大特点之一(参见图 5)。

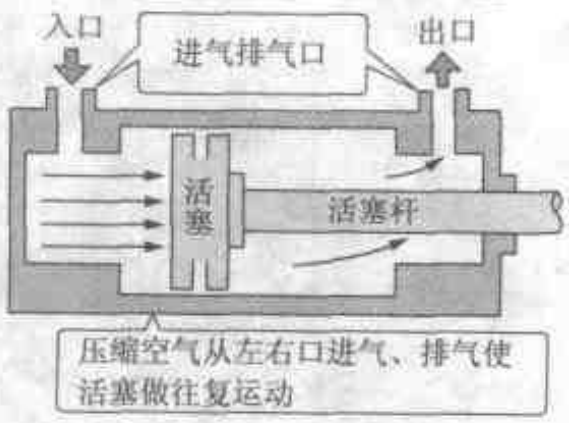


图 3 汽 缸



图 4 液体的不可压缩性

作为易于理解的油缸的例子，还可以举出自卸车的承载斗升降所用的油缸。

③ 螺线管

如图 6 所示，螺线管是利用电流流过空芯线圈时电流产生的磁场，使铁片（可动铁芯＝插棒）吸引到线圈中心。这时可动铁芯的动作是直线运动，图 6 的下端可利用其拉力而上端则可利用其推力。利用拉力的为拉力型螺线管，利用推力的为推力型螺线管，也有拉力与推力两用型螺线管。

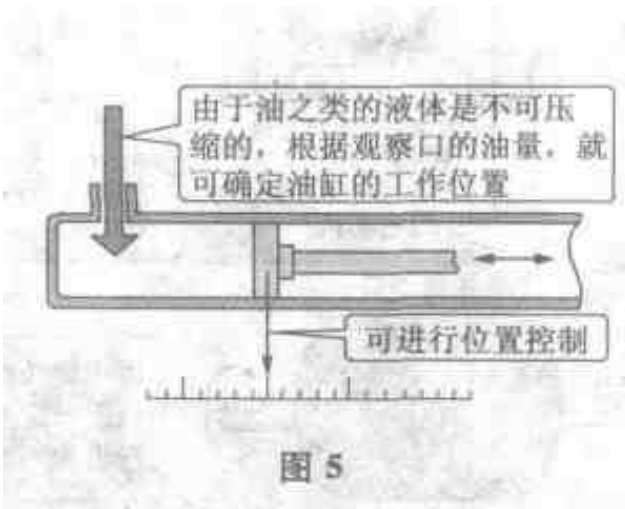


图 5

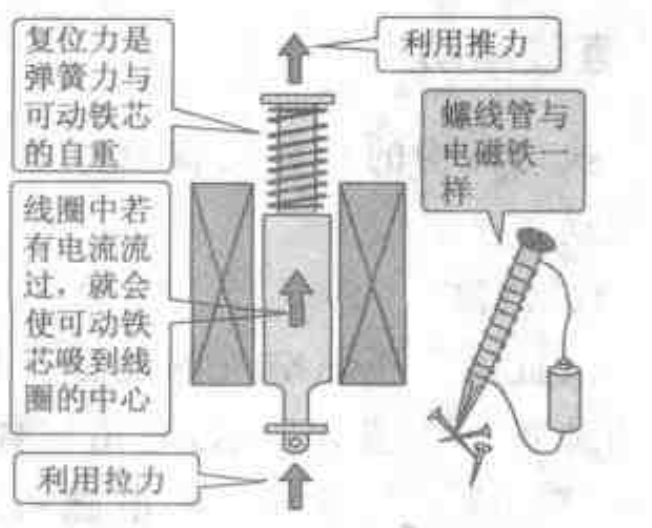


图 6　螺线管的驱动原理

螺线管也可用于前述的汽缸电磁阀的切换中。

2 执行机构的驱动电路

执行机构的驱动电路是变换控制信号所必需的电路，是为了将来自计算机的信号变换为使执行机构动作的控制信号的电路。

1 驱动执行机构用的电路

执行机构是不能直接用计算机的“1”、“0”信号驱动的。如图 1 所示，现在来考虑用一个来自传感器的输入信号，驱动执行机构的计算机控制系统。

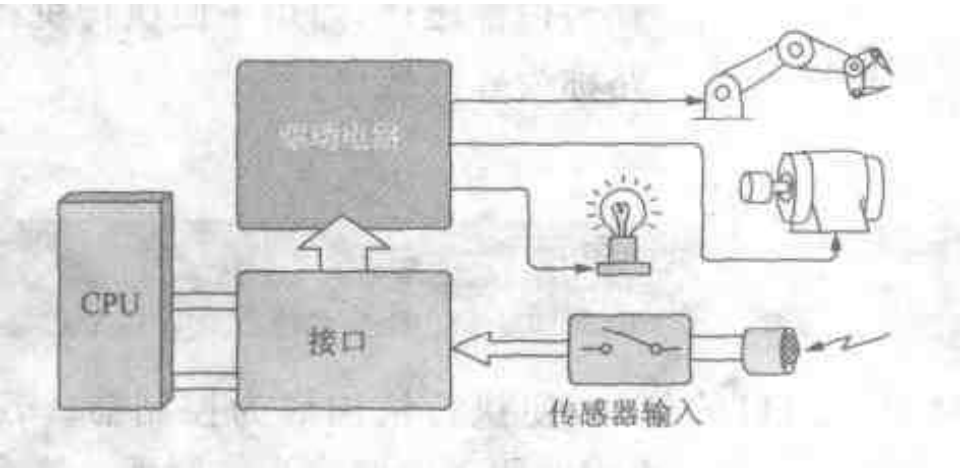


图 1　执行机构的驱动

驱动执行机构用的信号通过接口的输出口来输出的。如果将来自输出口的信号变换为使执行机构动作的信号，可以实现例如，图 1 中电灯点亮，作为执行机构的电动机的接通、切断，或使机械手夹住、松开等。为此必须用图 1 所示的执行机构驱动电路。

驱动电路是用来使执行机构动作的，所以也称为“传动电路”。图 2 所示为模型电动机与三相电动机的驱动电路。由图 2 可以知道，为什么需要各种各样电动机的驱动电路。

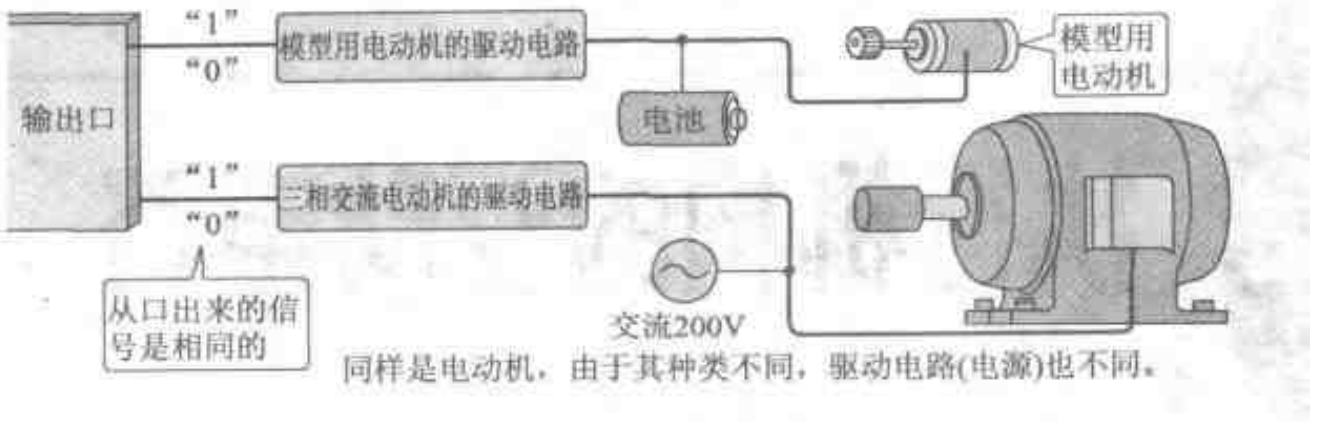


图 2　模型电动机与三相电动机的驱动电路

Ⅱ　接口的通用电路

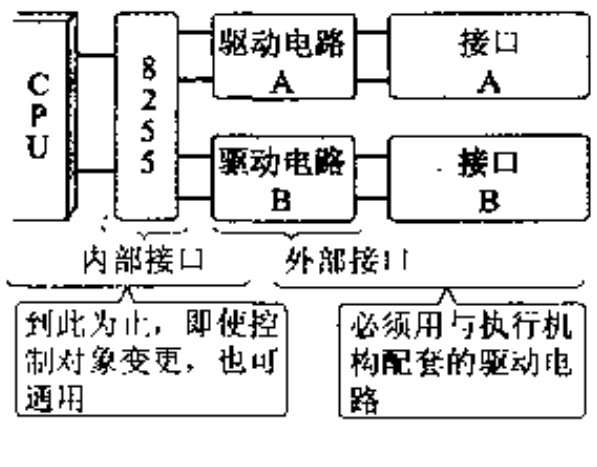


图 3　内部接口

由图 1 可知，到接口的输出口为止，基本上都是由几乎相同的电路构成的，而不用考虑后面接的是什么样的执行机构。因此如图 3 所示，通过将这一部分的通用(标准)化，就可用于其他的执行机构的驱动。

把这种能通用于各种各样输入输出的接口称为内部接口，而用于使执行机构动作的驱动电路称为外部接口。

Ⅲ　执行机构驱动用的信号

通过来自接口输出口的信号，使执行机构成为控制器。若说得更详细一点，是前面所说的信号使执行机构的驱动电路成为控制器。这个信号可分为三大类。

"1"、"0"的开关信号

计算机的"1"、"0"，其本身就是对应于执行机构驱动开关的接通、切断状态(参见图 4)。

脉冲信号

若将输出口的数字脉冲供给驱动电路，就得到与该脉冲成比例的动作。作为最容易理解的例子，如图 5 所示，是供给步进电机的脉冲数与轴的旋转角度的关系。

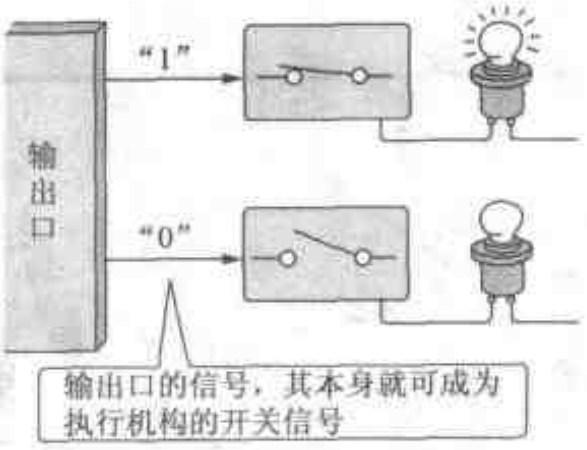


图 4 用"1"、"0"的信号驱动

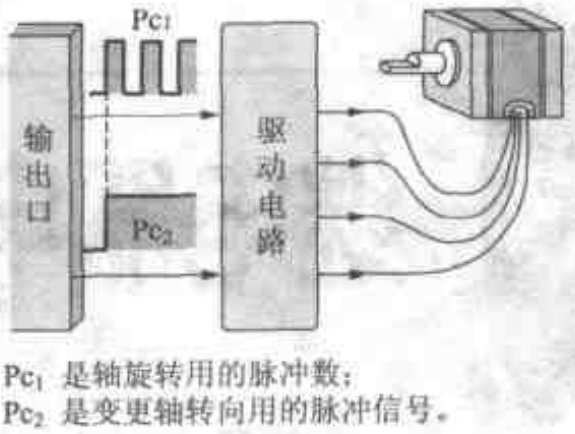


图 5 用脉冲信号驱动

来自数/模(D/A)转换器的信号

根据来自输出口的数据位数、输出对应的数字量，将这一数字量数据转换为对应大小的电压值，施加到执行机构上称为数/模转换。

图 6 所示为 8 位的数/模转换的作用。此数/模转换器用于直流电动机变速等场合。电动机的转速用旋转编码器检测，也有用反馈控制的。

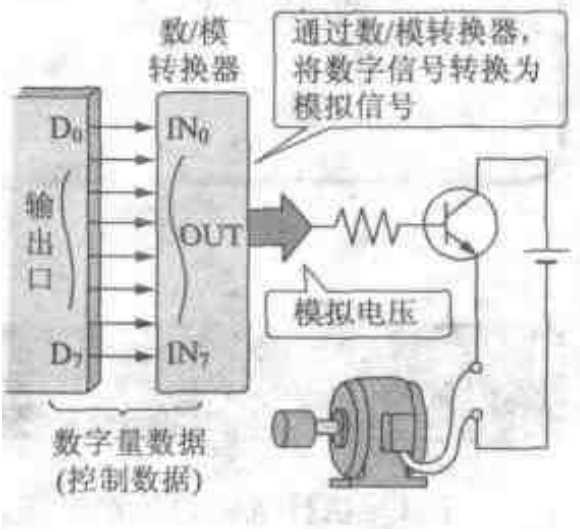


图 6 数/模转换的作用

3 继电器电路

继电器如右图所示，是利用在线圈(绕线)上流过的电流产生的磁力吸附铁的性质，用于电气接点作为开关的装置。也称为电磁继电器。



1 执行机构与继电器电路

表 1　执行机构的驱动电源

直流	5V	电流的大小不同
	12V	
	24V	
交流	100V	
	200V	

仅仅用计算机输出信号本身来使电动机或汽缸等执行机械动作，其力量是不够的。所以常使用继电器以接口信号控制线圈电流控制外部电源。依靠这种继电器，就能控制不同电源种类的执行机构。表 1

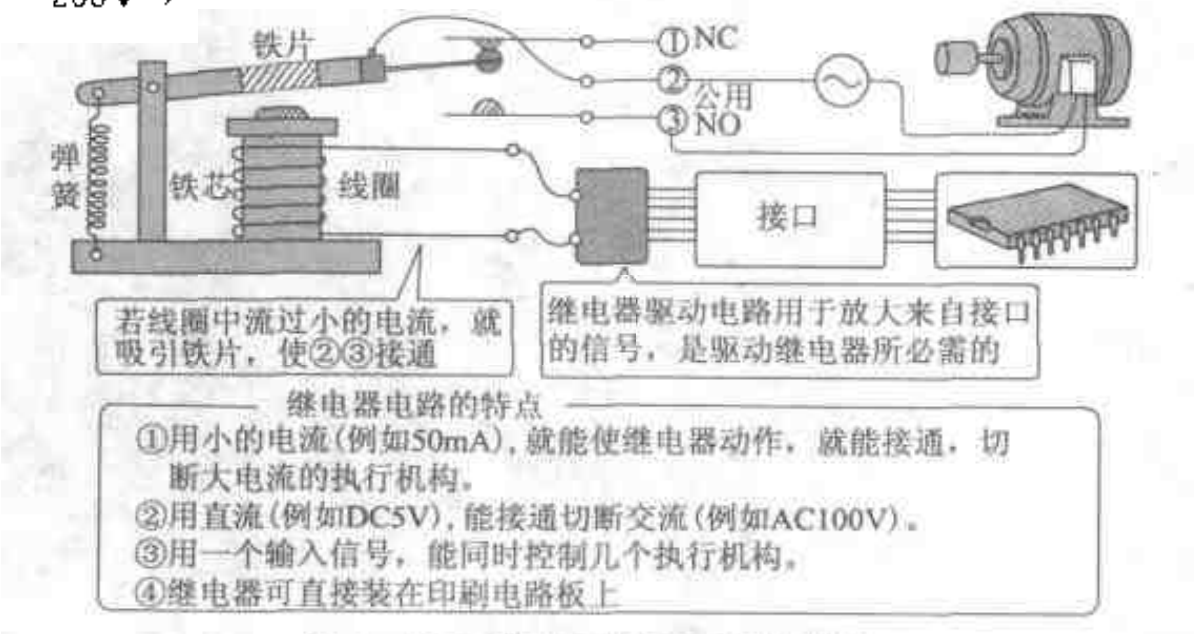


图 1　继电器驱动的原理及其特点

表示常用的执行机构的驱动电源种类。

继电器的重要用途之一，就是用在接口的控制上。这类用途中，机械接点继电器的原理图及特点如图 1 所示。将图 1 的执行机构电动机，换成其他类型的执行机构，比如汽缸或螺线管等，也能对执行机构进行接通与切断的控制。

继电器电路是驱动这类执行机构用的非常方便的电路。

小灯泡的亮、熄

图 2 是用继电器驱动两个执行机构（这里是小灯泡亮、熄的例子）的原理图。通过来自计算机的信号，可确认切换的操作。

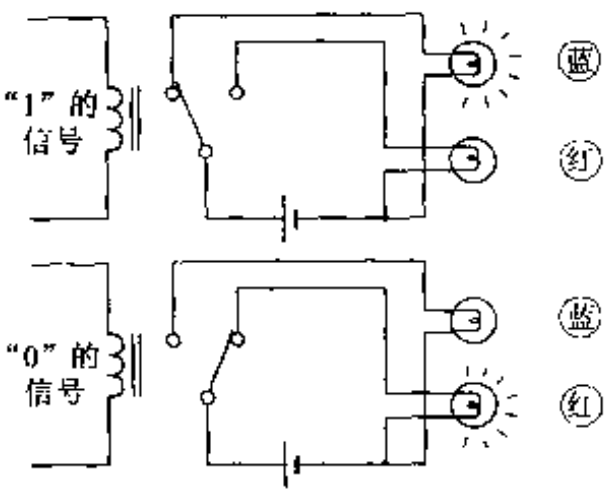


图 2 小灯泡的继电器控制

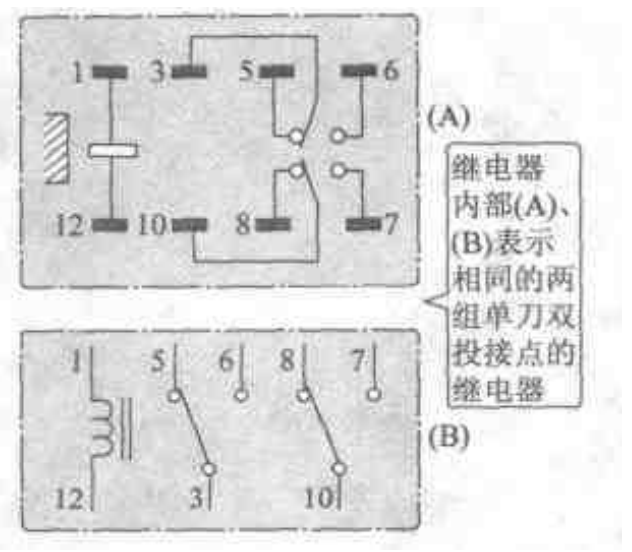


图 3 继电器内部电路

图 3 表示的电路是常用于印制电路基板上的继电器的构造。这是一个称为带两组单刀双投接点的继电器，若图 3 的引脚 1 与 12 之间输入了来自计算机的信号，则线圈通电励磁，原来引脚 3 与 5 及 10 与 8 导通的接点，就切换为 3 与 6 及 10 与 7 导通。这种带两组单刀双投接点的继电器，使用一个信号，可同时控制两套执行机构。

制造厂提供各种不同构造的继电器，可查阅制造厂的产品样本。以前所介绍都是有接点的继电器，另外还有集成（IC）化的无接点继电器。

继电器的应用

图 4 所示为使用继电器的具体应用例子。

① 通过两个接点来切换施加在直流电动机上的电源极性，就能改变旋转

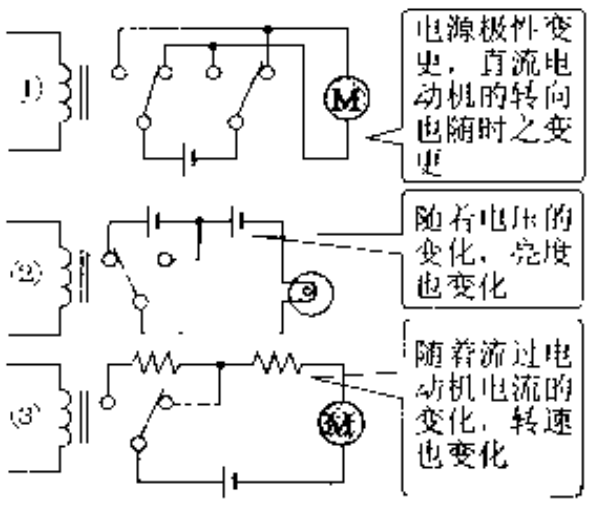


图 4　继电器的应用电路

方向。

② 切换灯泡上所施加的电压，就能改变灯泡的亮度。

③ 通过改变流过电动机电流的大小，可改变电动机的转速。

控制用的继电器，应当选用易于在印制电路基板上插装的适合 IC 管脚距(2.5mm)的继电器。

4 使用晶体管的驱动电路

1 用晶体管驱动执行机构

前面已叙述过关于用继电器来驱动的执行机构。但实际上，来自计算机的信号，要用于驱动如图 1 所示的继电器，力量是不够的。所以必须用晶体管的电流放大作用，构成继电器的驱动电路，来获得驱动继电器所需的足够电流。

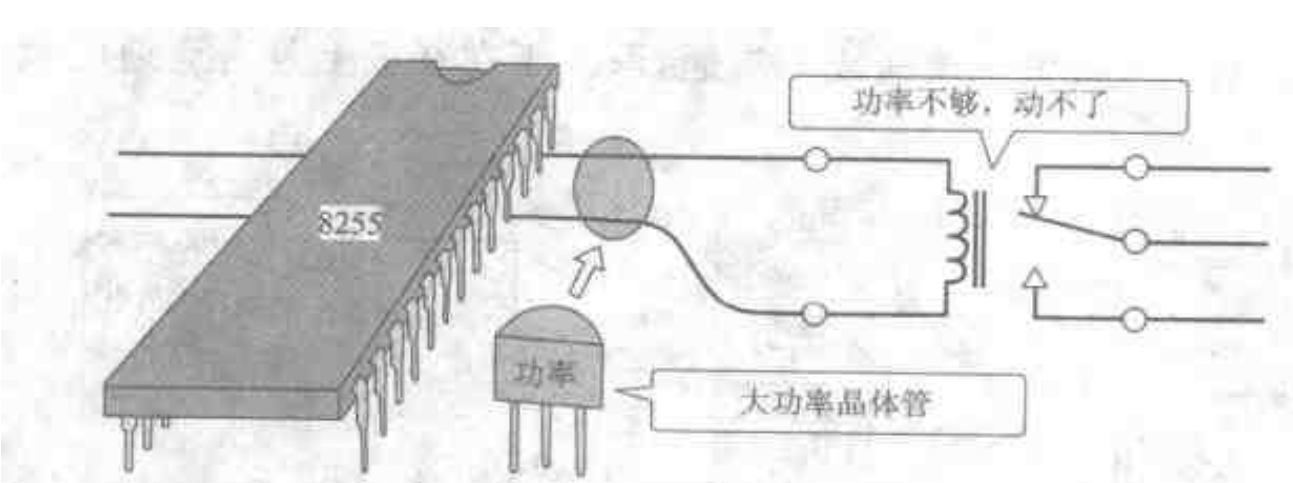


图 1　从 8255 输出的信号的功率

电流放大作用的定义是利用晶体管的性能，将小信号电流放大，变成大信号电流。如图 2 所示，若基极(B)与发射极(E)之间流过来自计算机的微弱电流，则集电极(C)与发射极(E)之间就流过大的电流，使继电器线圈励磁，能使继电器接通(ON)。

利用晶体管的这一放大作用，就能直接驱动执行机构。

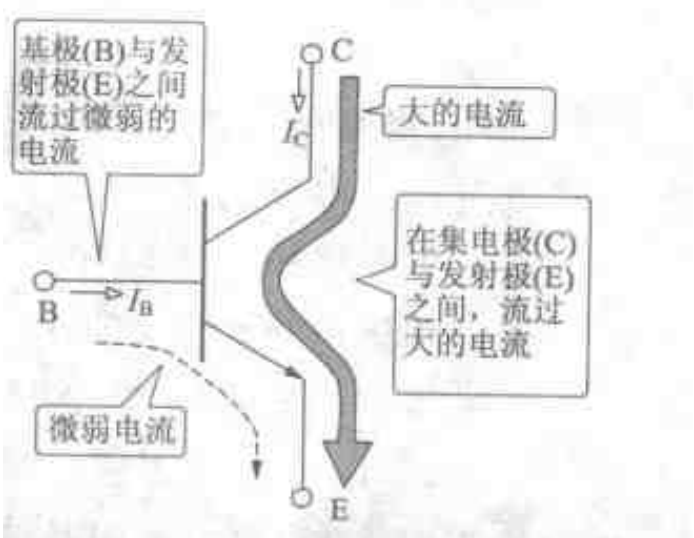


图 2　流过晶体管的电流

Ⅱ　晶体管的电流放大系数

如图 2 所示，基极上流过的小电流，会产生集电极流向发射极的大电流。此时，经放大的集电极电流 I_C 与基极电流 I_B 之比值，就称为电流放大系数。

$$h_{FE}=\frac{I_C}{I_B}$$

根据晶体管的不同，这一电流放大系数可从数十倍到近千倍。

在图 2 中，当流经基极的电流为 0.02mA 时，若要流过集电极的电流为 2mA，则此时的电流放大倍数如下式所示。放大后的集电极电流，就是流入继电器或执行机构的电流。

$$h_{FE}=\frac{I_C}{I_B}=\frac{2}{0.02}=100$$

Ⅲ　晶体管的开关状态

被放大的集电极电流流通或不流通，取决于在基极上是否流通电流（参见图 3）。

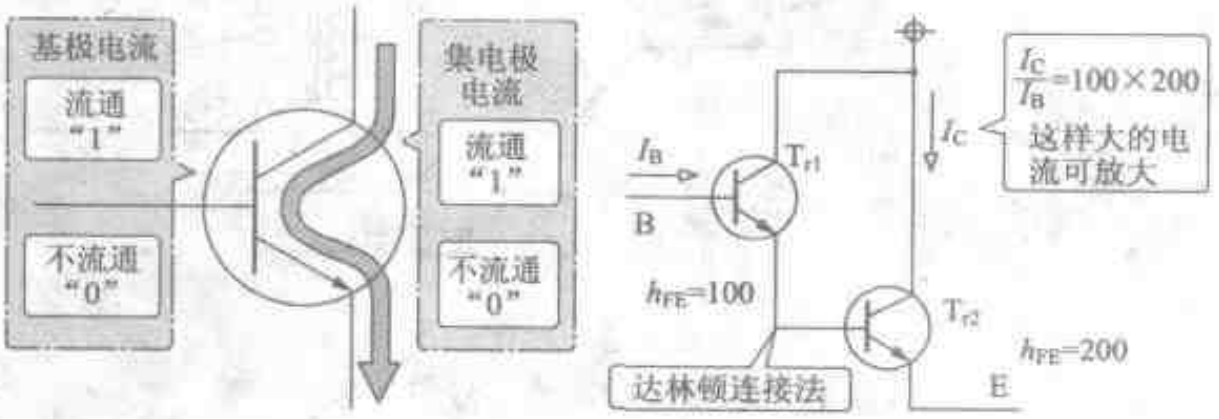


图 3　晶体管的开关状态

现在，将基极电流假定为数字量“0”、“1”的值，

流过基极电流为零“0”，

流过集电极电流为零“0”，

流过基极电流为 0.02mA“1”，

流过集电极电流为 2mA“1”。

这样，可以通过检测流过基极的电流，就知道流过集电极的电流，将这一晶体管的工作状态，称为利用晶体管的开关状态。

Ⅳ　晶体管的选择

一般情况下，晶体管的选择顺序可以按图 4 所示的流程来考虑。

不同类型的晶体管，有其保证正常工作的最大额定值。应当选择能适合于执行机构和继电器负载的、最大额定值足够大的晶体管。通常，应选择晶体管的额定值为所需电压与电流值的 2～3 倍。实际工作中常通过查表来决定。

选择好晶体管后，通过该晶体管的电流放大系数可以核查从计算机得到的基极电流是多大，若基极电流不适当，则即使来自接口的信号输出为正常值，也不能工作。在这种情况下，如有必要，可选择电流放大系数更大的晶体管，或如图 3 所示用达林顿接法连接，来得到合适的基极电流。

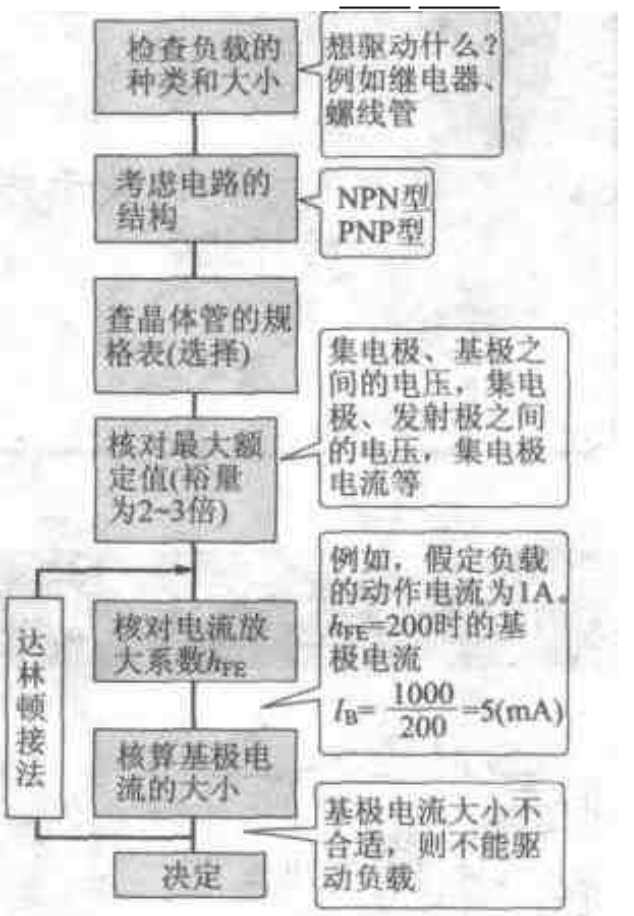


图 4　晶体管的选择顺序

5 螺线管驱动电路

介绍关于作为直线运动执行机构的螺线管。



Ⅰ 螺线管的动作原理

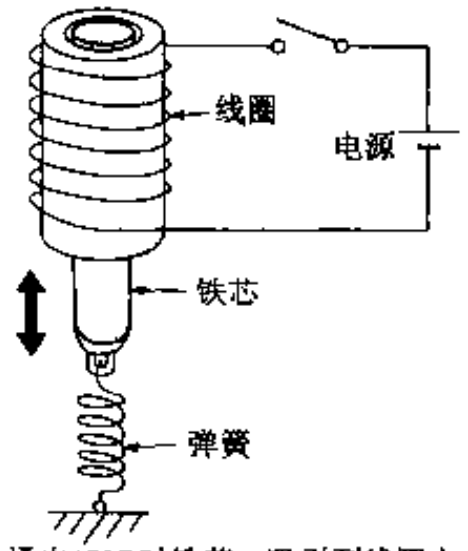


图 1　螺线管的动作原理

如图 1 所示，螺线管是靠线圈中流过的电流产生的磁力来吸引可动铁芯的执行机构。比如，要将物体推开、拉动、压住、打开等单纯机构的机械直线运动部件，都可以利用这种执行机构。另外，自动售货机，办公设备或家用电器等，也装有直接利用螺线管作为直线运动的部件。

螺线管的驱动类型，有直流型与交流型。其驱动的额定电压一般为

直流：5V、6V、12V、24V、48V

交流：100V、200V

具有同样大小吸力的螺线管，其电压越大、线圈中流过的电流越小。

Ⅱ 螺线管的使用要点

吸　力

吸力是螺线管定额中最重要的指标。若吸力不够，不仅螺线管的功能不能

发挥，而且还可能使其他的机械结构部件遭到破坏。负载的大小必须小于螺线管的吸引力。

在使用状态中，由于电压的降低与温度的上升，螺线管的吸引力也会下降，因此制造厂推荐的值是以85%额定电压时的吸力为标准的。这一吸力在样本中是以"N"(牛[顿])为单位表示的，它与力的另一单位"kgf"的关系如图2所示。螺线管吸引力的大小，从2.94N(0.3kgf)到9.81N(10kgf)的都有。

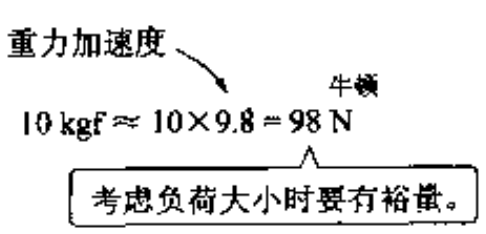


图2　力的单位的换算

另外，必须注意相对于额定的吸力，空载状态的"空击"，或负载在30%以下的轻载的使用，都会缩短螺线管的使用寿命。

行　程

选择螺线管时，行程大小也是重要的指标。如图3所示，行程加大，则吸力就急剧减小。因此，在额定行程以内，也应使用尽可能短的行程。

若行程加长，使铁芯不能正常地返回的话，则在线圈中流过大电流，这是造成故障的原因。在需要长行程的情况下，如图4所示，以杠杆或汽缸等机械机构组合使用为好。

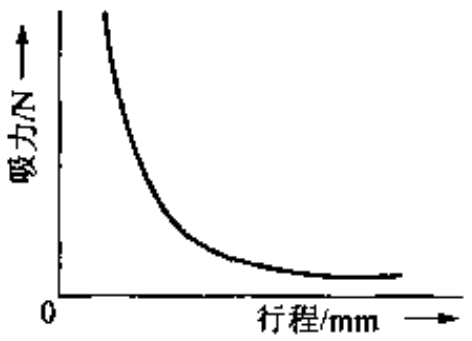


图3　吸力与铁芯行程的关系

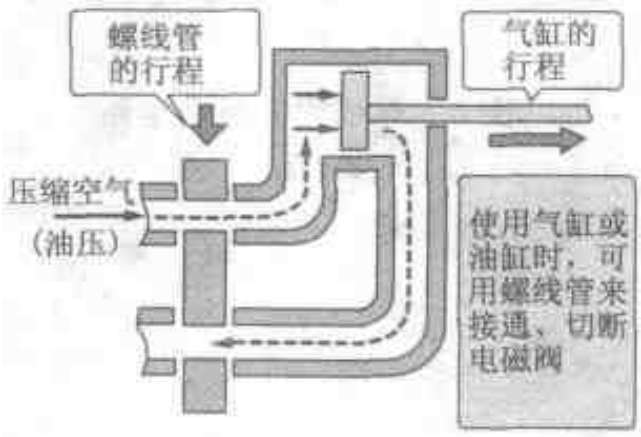


图4　与汽缸组合的使用

将动铁芯紧贴在固定铁芯上

如图5所示，在吸引时若动铁芯不能紧贴在固定铁芯上，也会增大线圈中的电流，成为损坏的原因。因此，在安装中必须使固定铁芯与动铁芯的接触面完全地接触。

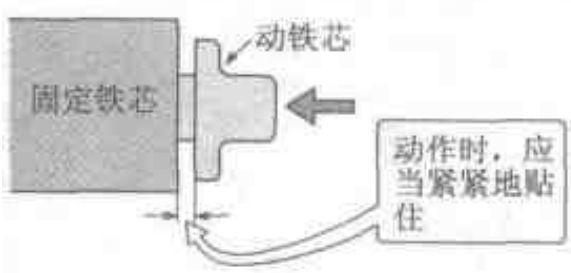


图5　固定铁芯与动铁芯的隔离

Ⅲ　螺线管的驱动电路

图 6 所示为使用执行机构与螺线管驱动电路的例子。通过 8255 输出的信号，使继电器动作，驱动螺线管。

在此电路中，通过继电器电路，可由外部电源来提供与螺线管工作电压相当的电源电压。在此电路中使用欧姆龙公司的继电器(G2VN-237P)，因其接点容量可达 3A，所以此范围内的螺线管都能驱动。

另外，在切断螺线管时，由于反电势造成的高电压，很可能损坏晶体管与集成电路。作为保护措施，如图 7 所示，在螺线管的线圈上并联接入二极管。

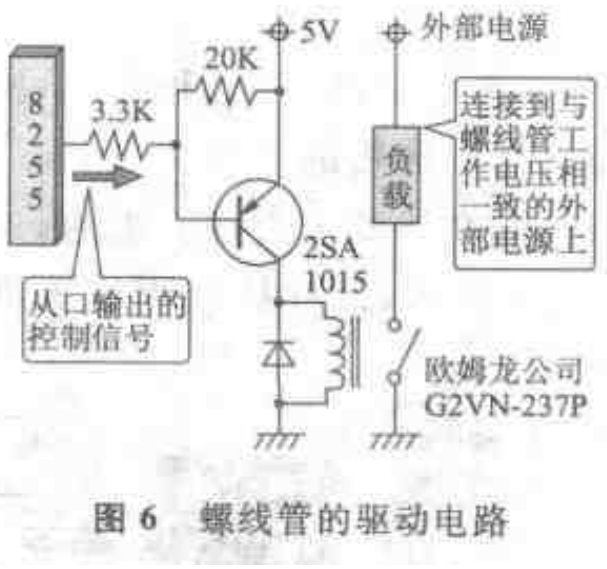


图 6　螺线管的驱动电路

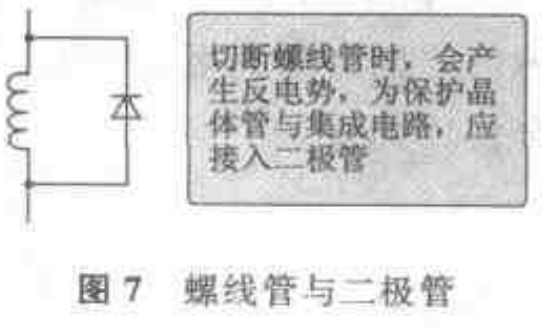


图 7　螺线管与二极管

6 步进电动机的驱动电路

Ⅰ 步进电动机

步进电动机与普通电机不同，仅在线圈上施加电压是不会旋转的。

步进电动机是通过施加脉冲，根据脉冲数来旋转的电动机。因此，利用脉冲数量，可以准确地控制旋转角度。比如，如图 1 所示，若发送 100 个脉冲时，电机轴的旋转角为 180°，则发送 200 脉冲时的旋转角度就是 360°。

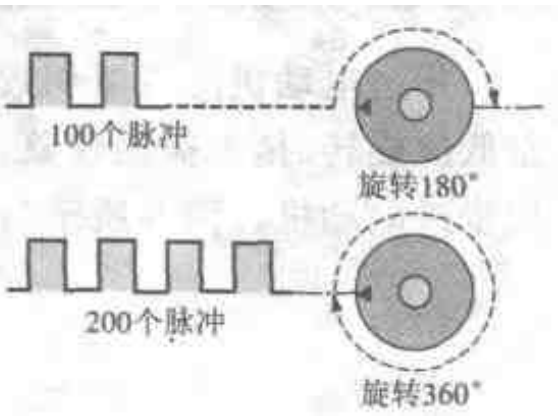


图 1　脉冲数与旋转角度成比例

这样，不但能通过传感器的信号，而且也能通过计算机的输出信号来准确地控制旋转角度。也就是说，步进电动机是适合计算机控制的电动机。

作为我们熟悉的例子，打印机打印头驱动用的电动机，就是步进电动机。

在文字处理等印刷机中，竟能如此精密、令人佩服地打印，也是步进电动机的功劳。

Ⅱ 步进电动机的原理

当图 2 所示的线圈 A 中流过电流时，定子变成 N 极，就吸动转轴（转子）的 S 极。接着，切断线圈 A 的电流，若使电流流过线圈 B，则转子的 S 极，就被吸到线圈 B 所产生的 N 极。这样，就随着励磁线圈的通电，而使转子连续地旋转。

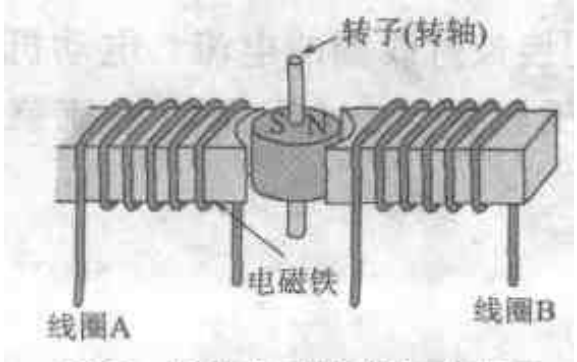


图 2　步进电动机的驱动原理

Ⅲ 步进电动机的特点

① 脉冲数与转角成准确的比例；

② 每一步所对应的角度误差小；

③ 电动机的正反转、停止响应快捷；

④ 能用数字信号的输出脉冲来控制(能开环控制)；

⑤ 只要施加电压，步进电动机就有自锁(制动)力，因此能保持在停止位置。

由于步进电动机有这些特点，所以可用于计算机控制中。

Ⅳ 步进电动机的驱动电路

步进电动机的线圈相数，有双相、三相、四相、五相等。步进电动机的缺点是低速旋转时，有振动现象，为了得到更小的步进角分辨率，制造厂推荐使用五相步进电动机。图3所示为常用的四相步进电动机的基本驱动结构。

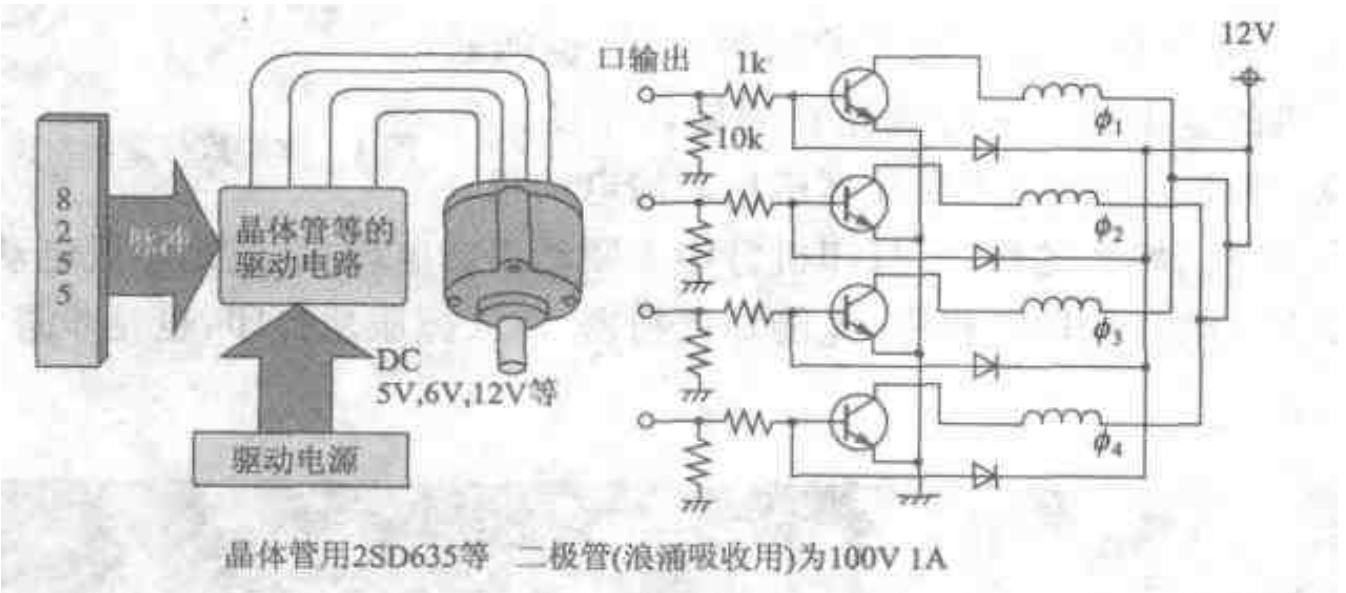


图3 基本驱动结构图

由计算机来提供脉冲，再通过晶体管开关来切换流过线圈的电流。电动机驱动常用的电源为直流5V、6V、12V。额定电压越大，流过线圈的电流就越小。

Ⅴ 步进电动机的驱动

如图 4 所示，除了晶体管步进电动机驱动电路外，在 IC 型封装中还包含了四个晶体管的驱动电路，作为步进电动机驱动器（晶体管阵列）。使用这种晶体管阵列可简化驱动电路。例如，用三菱电机公司的晶体管阵列 M54532P。

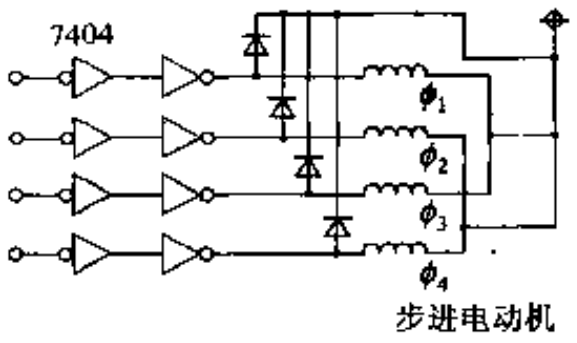


图 4 三菱 M5432P 驱动器

Ⅵ 励磁方式

对四相步进电动机提供驱动脉冲的方法，有以下三种。

① 单相励磁。四相中，经常只有一相流过步进脉冲电流。

② 双相励磁。四相中，经常有两相流过步进脉冲电流。

③ 单-双相励磁。单相励磁与双相励磁交替进行。步进角度为 1/2，适用于要将步进角分得更小的场合。

图 5 表示由双相励磁提供驱动脉冲的方法与数据。要使步进电动机改变转向，只要通过程序，将提供脉冲数据的顺序倒过来就行了。

	双相励磁方式				数据	
ϕ_1	0	1	1	0	&H6	↑
ϕ_2	0	0	1	1	&H3	
ϕ_3	1	0	0	1	&H9	
ϕ_4	1	1	0	0	&HC	↓

(&H表示十六进数) 正转 反转

图 5 双相励磁方式的驱动数据

7 固体继电器的驱动电路

家用电器用的100V电源可以利用计算机的信号来控制，其应用范围在不断地扩大。

这里介绍一下关于常常作为半导体继电器用的固体继电器。

I 交流100V电源的控制

固体继电器有直流控制用与交流控制用两种之分，但一般把交流控制用的称为SSR。图1表示SSR的外观。输入引脚(INPUT)上施加操作电压(5V、12V、24V)，将负荷电压(例如交流75～250V)连接在输出引脚(LOAD)上。

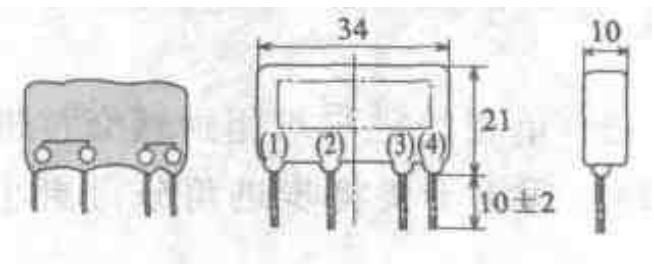


图1 SSR的外观形状

从用途上看，SSR可与机械的电磁继电器同样使用。其形状紧凑，所以电路构造简单。

II SSR的内部构造

以前所介绍的继电器，都是由电磁线圈为输入端，接点电路为输出端组成的。而SSR则如图2所示，是使用光电耦合器，用光来传递信号的。因此，能完全隔离接通、切断负荷时所产生的噪声等。

SSR的动作是通过输入端的信号，使发光二极管上流过电流而产生光，该光使光晶体管导通，就能控制输出电路。

过零反转电路的定义是(参见图3)，接入信号时，交流负载端在零电位附

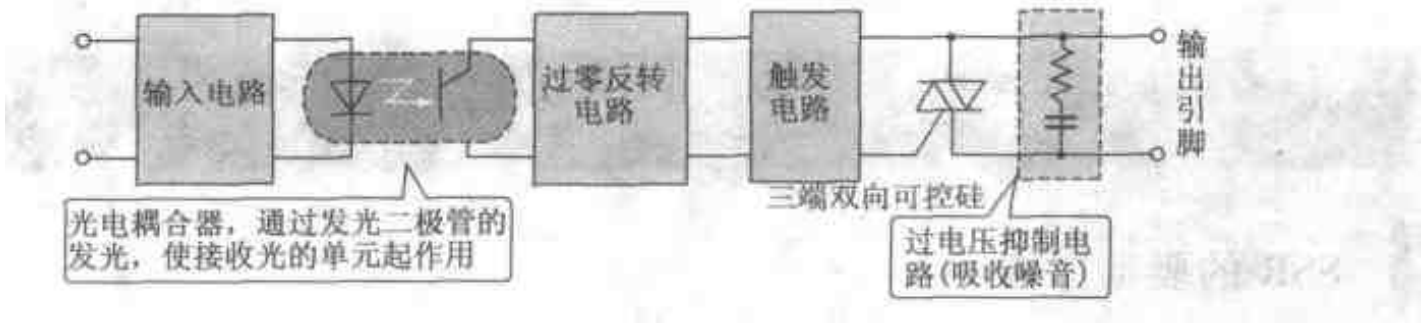


图 2　SSR 的信号传递

近触发，使三端双向可控硅导通；切断时，也是在负载电流到零附近触发切断。因此可减少伴随着控制信号产生的噪声或浪涌电压。

连接螺线管或电动机等的感性负载时，由于电流对电压的相位滞后，会在 SSR 上施加浪涌电压。过电压抑制电路即是为此而设的保护电路。

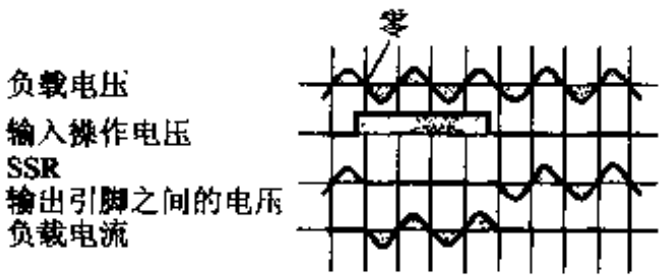


图 3　过零反转电路的动作

Ⅲ　使用 SSR 时的注意事项

SSR 可通过来自计算机的信号直接控制交流电压 100V、200V 的电路，因此在交流执行机构控制中，是利用价值很高的元件。在使用中应注意以下事项：

① 当交流负载端有大的浪涌电压影响时，为了防止误动作，应当连接压敏电阻(参见图 4)。

② 对小容量的负载来说，应当接入与负载并联的假负载电阻(参见图 5)。

③ 确定所使用的输入信号管脚(＋、－极性)，不得搞错。

④ 应注意 SSR 的使用环境温度。

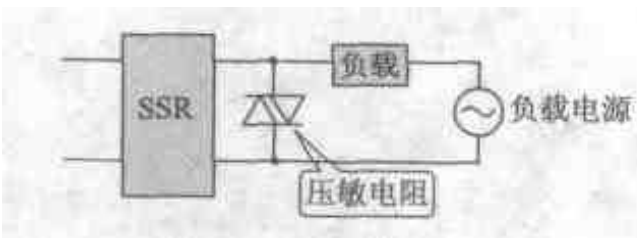


图 4　压敏电阻的连接



图 5　小负载情况下假负载电阻的连接

Ⅳ　SSR 驱动电路的例子

SSR 的驱动方法

图 6 所示为 SSR 的驱动方法的例子。

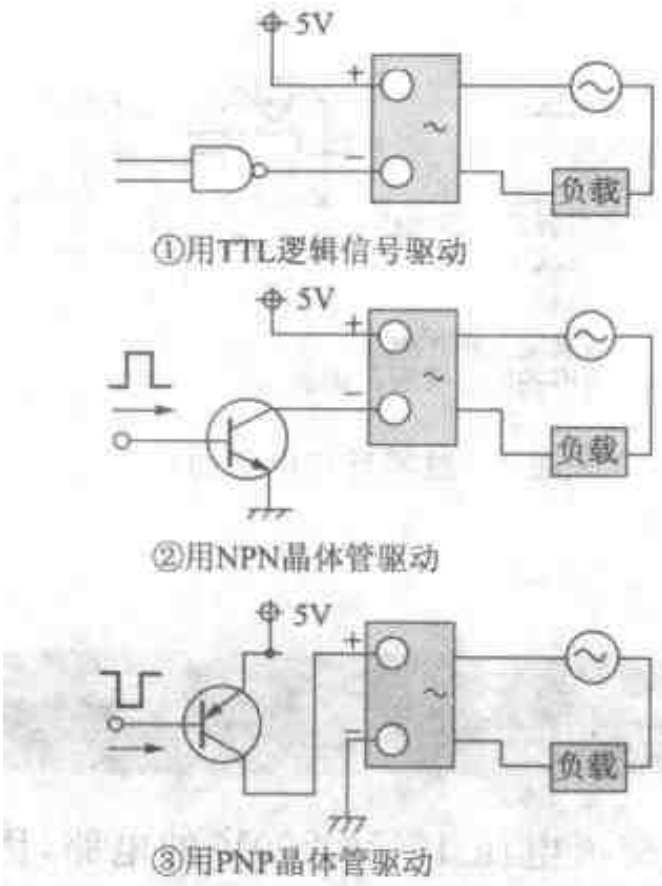


图 6　SSR 的驱动方法

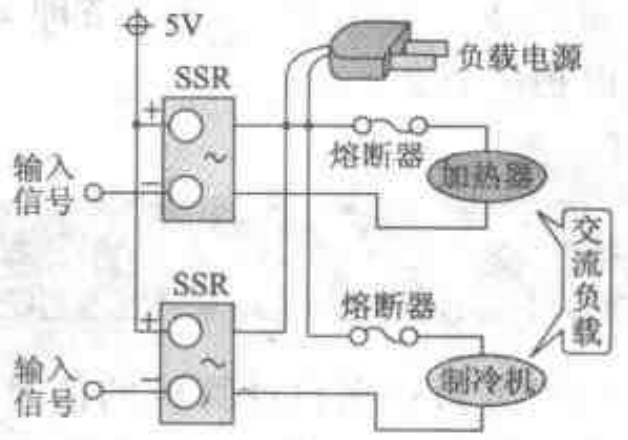


图 7　交流负载电路

利用 SSR 的电路例子

图 7 所示为利用 SSR 的交流负载电路例子。若在负载中连接电磁继电器，则可以连接大功率的负载。如图 7 所示，交流负载电路中接入了熔断器。

Q 挑战题

① 执行机构是什么?
② 执行机构的两个基本运动是什么运动?
③ 要使汽缸的气流方向切换,用什么装置?
④ 与汽缸相比,油缸有何特点?
⑤ 称执行机构的驱动电路为什么?
⑥ 什么是内部接口?
⑦ 把数字量数据转换为模拟量数据的叫作什么?
⑧ 继电器的驱动电源有几种?
⑨ 利用半导体 IC 取代机械接点的继电器叫作什么?
⑩ 什么是晶体管的达林顿电路?
⑪ 描述一下螺线管的工作原理?
⑫ 写出表示力的单位"N"与"kgf"的换算公式。
⑬ 螺线管的吸力与行程有何关系?
⑭ 螺线管断开时,产生的电势叫作什么?
⑮ 对⑭题中的电势,有什么对策?
⑯ 用每一个脉冲旋转 1.8°的步进电动机来旋转 90°,需要多少个脉冲?
⑰ 使步进电动机反转,要提供什么样的脉冲?
⑱ 步进电动机的励磁方式有几种?
⑲ SSR 是什么的缩写?
⑳ 过零反转电路的定义是什么?
㉑ SSR 中用什么传递输入信号?

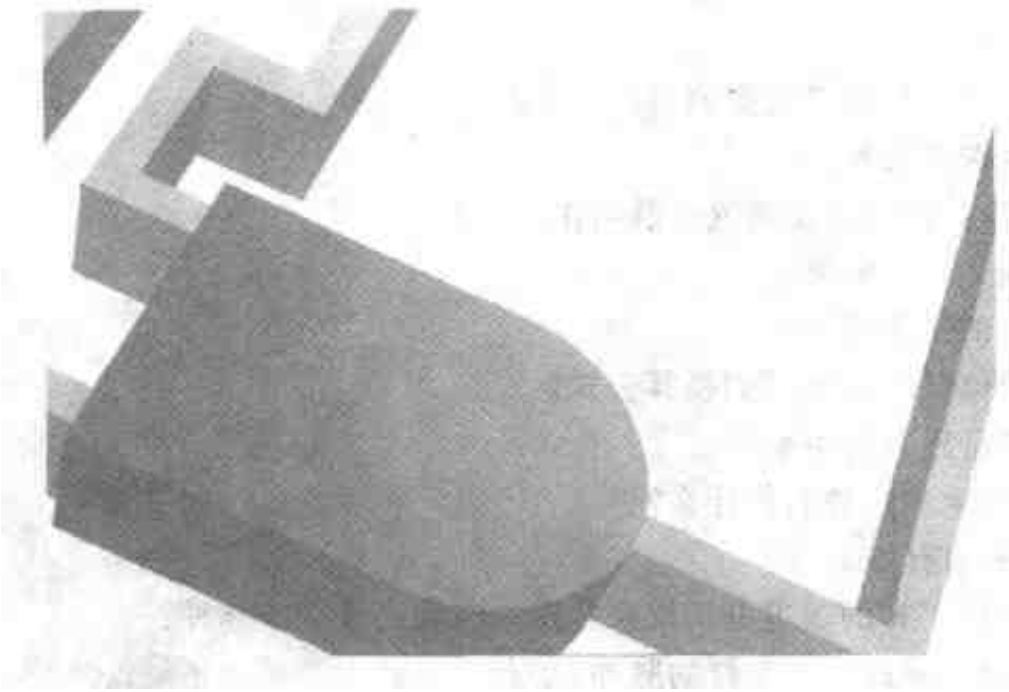


A 答案

①略,参见第 1 节。 ②略,参见第 2 节。 ③螺线管。 ④液压力的传递媒介是液体,因为是非压缩性的,因此可用于位置控制。 ⑤传动电路。 ⑥略,参见第 2 节。 ⑦数模转换。 ⑧DC5V、6V、12V;AC100V、200V。 ⑨无接点继电器。 ⑩参见第 4 节。 ⑪略,参见第 5 节。 ⑫10(kgf) =98(N)。 ⑬行程加大则吸力急剧减小。 ⑭反电势。 ⑮在线圈上并联二极管。 ⑯50 个脉冲。 ⑰供给与正转相反顺序的脉冲。 ⑱单相,双相,单-双相。 ⑲固体继电器。 ⑳略,参见第 7 节。 ㉑光电耦合器。

第4章

控制的基础知识



控制与逻辑电路的基础

“控制”这个话题，有时候让人望而生畏，这样的关于控制的学习，我想应尽量避开。

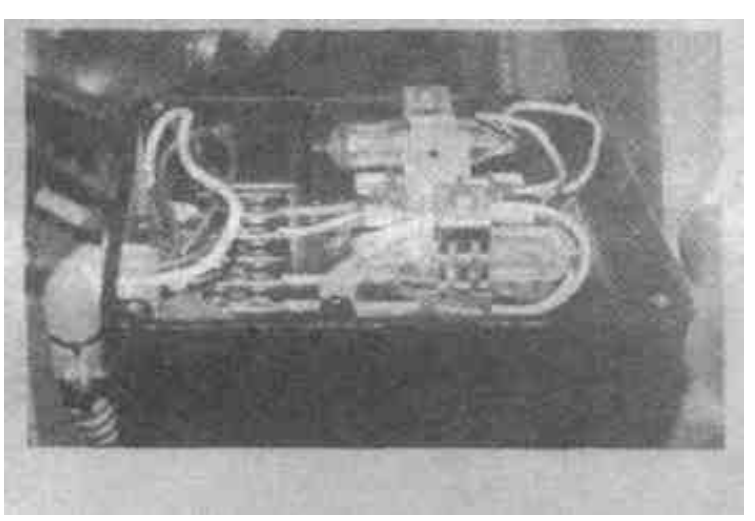
然而，自动化不但进入了我们身边的洗衣机等家电产品，也深入到了工厂、办公室中，如工厂自动化(FA：Factory Automation)，办公室自动化(OA：Office Automation)等。模糊控制、微机控制的话题也常被人谈起。

就当前的信息化时代而言，都要求以时代的需要与目标来开发新产品，若要有效地灵活使用这些机器，则必须掌握控制技术。

控制技术进一步发展为神奇的黑匣子，通过它可有效地灵活使用这些机器，并能最大限度地发挥其价值与功能，真可谓妙极了。

由此可见，学习控制技术的基础知识是非常重要的。

在这一章中，我们的做法是边制作试验电路，边学习“控制”知识，也就是学习所谓难以掌握的控制知识或逻辑电路等基础知识。



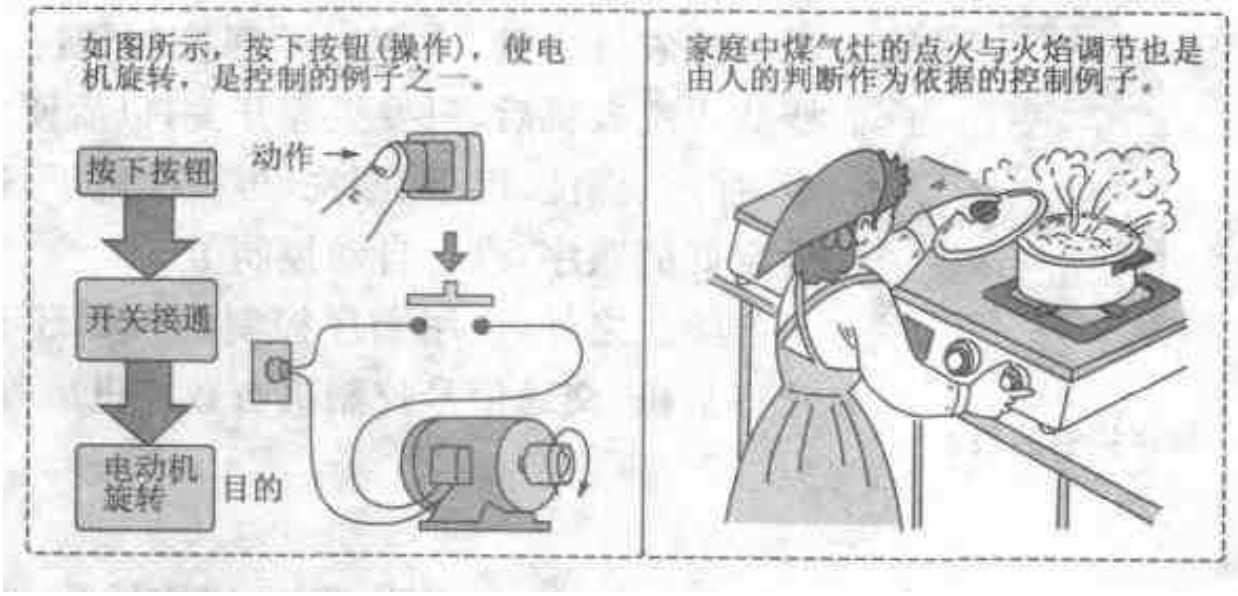
在开始阶段，学习基本理论是非常重要的，因此，大家务必在这一章的制作IC逻辑实验板的同时，学习控制的基础理论知识。

我想，IC逻辑板可对开始学习控制技术的人，在理解逻辑电路时起重要的作用。

1 控制的定义

首先我们探讨一下“控制”是什么概念。

日本标准 JIS 将控制定义为“为了达到某一合适的目的，对作为对象之物施加所必要的操作”。我们考察一下身边的例子。通过考察，可见在我们的周围控制现象到处存在。



Ⅰ 自动控制的定义

以电冰箱的温度控制为例进行探讨。

如果冷冻箱内温度升高，就自动地接通制冷电动机的开关制冷，太冷了就自动停止。冷冻箱内的开关动作如图 1 所示，是通过称为热敏电阻的控制器件，自动地、反复地进行接通、切断操作的。像这样通过控制装置，自动地进行的控制称为自动控制。

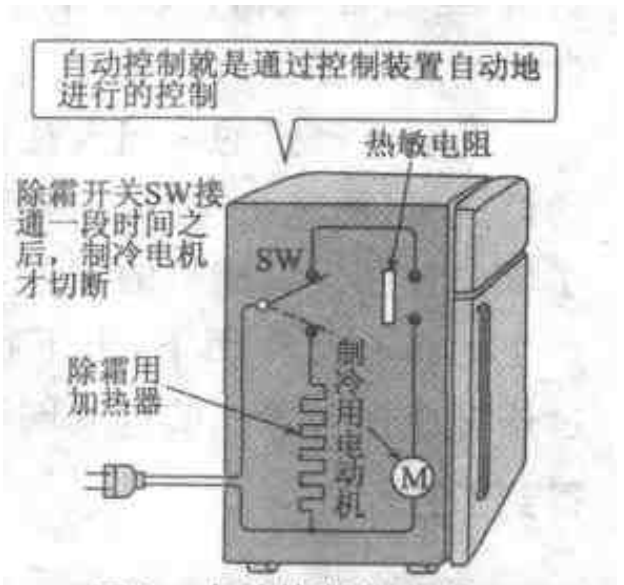


图 1　自动控制的定义

热敏电阻例如双金属片[1]，是利用液体或金属的膨胀变化，使接点断开与闭合功能的温度控制元件，已用于电暖器、电熨斗的温度控制中。

Ⅱ　控制的方式

控制的方式可分为两大类：控制系统呈开放状态的顺序控制与控制系统呈闭合状态的反馈控制。

顺序控制



图 2　顺序控制的例子

所说的顺序(sequence)就是英语按次序的意思。顺序控制是“根据事先编制好的顺序，依次一步步地操作所进行的控制”。

如图 2 所示，以事先准备好水、电、洗涤剂之后的洗衣机为例，来探讨顺序控制情况。把脏衣服放入洗衣机后，只要接通开关，以后便如图 2 所示，加水、漂洗、甩干、漂洗、再一次甩干，就按预先设定好的顺序，进行自动控制了。

除此之外，使用顺序控制的机器还有自动果汁售货机，交通信号控制机与数控机床等。

反馈控制

反馈控制是“将装置或机械的某一量(控制量)与目标量相比较，对所得的差值反复地自动地进行修正的控制。”

发明蒸汽机的英国人瓦特，使用图 3 所示的调速机(稳速机构)来运行蒸气机，就是使用了反馈控制。调速机的摆球，在蒸汽机转速所产生的离心力作用下，上下摆动，它的动作通过连杆机构与蒸汽的进气阀连动，就使蒸汽量得到了控制。

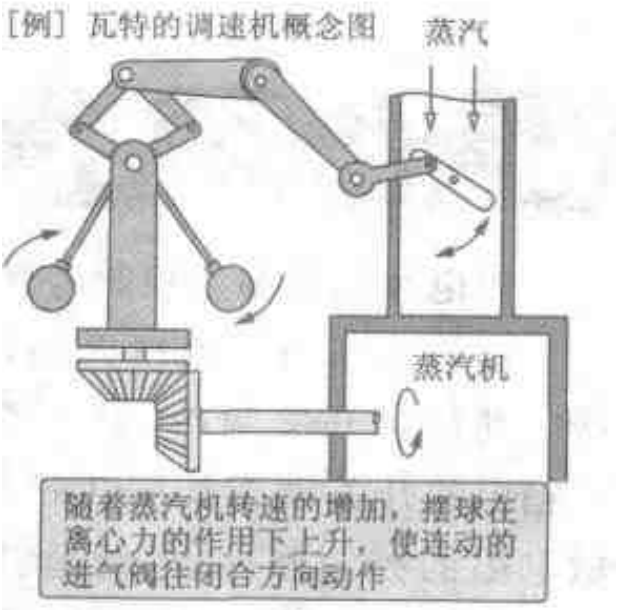


图 3　反馈控制的例子

1) 双金属片：由两片热膨胀系数不同的金属片组成，由于温度变化，内部变形不同，而使金属片发生弯曲。

举 例

① 顺序控制的例子

如自动售货机、电梯等。这类控制都是根据事先设定的顺序，依次按步骤进行操作的控制。

② 反馈控制的例子

如电炉的温度控制、水箱的供水等。将机械或装置等动作的数据量（把这称为控制量）与目标量相比较，反复、自动地修正其产生的差值所进行的控制动作。

2 计算机控制的定义

1 计算机控制

前述的“自动控制”这一术语所代表的顺序控制或反馈控制所需的控制部分，都是由继电器电路或集成电路等强电、电子电路构成的，将计算机功能加到此控制部分上，就是计算机控制。

现在来讨论一下图 1 所示的电灯 1 与电灯 2 的交替亮灯、熄灯控制。用电子电路对电灯进行交替亮灯、熄灯控制，是用定时器来控制亮灯时间，用继电器来进行开关切换的。

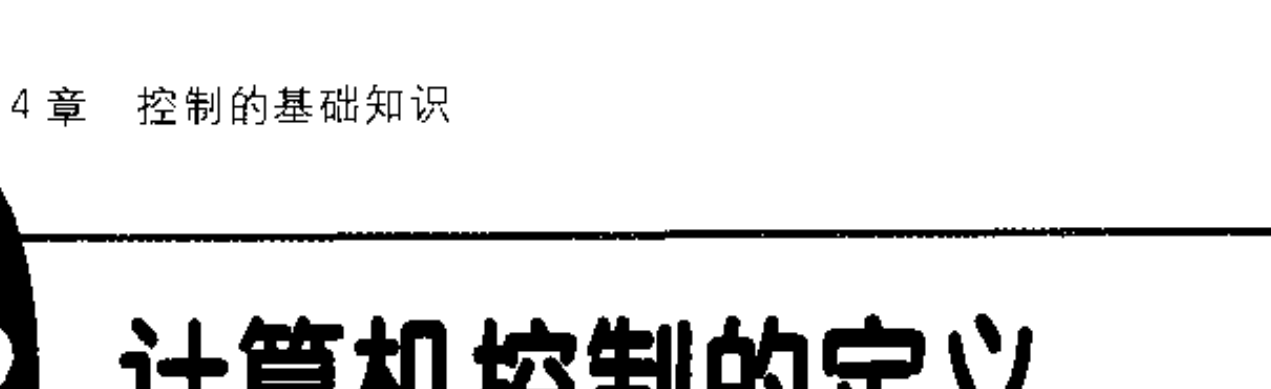


图 1　用电子电路控制

若使用计算机，则如图 2 所示，开关的动作与定时器都被计算机取代，因此

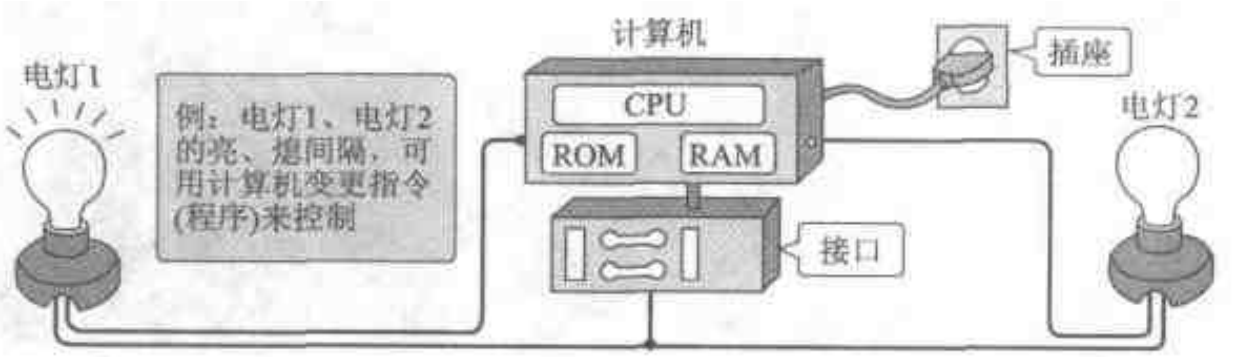


图 2　用计算机控制

通过程序，就可以决定某电灯的亮灯与否及亮灯的时间。像这样以计算机控制目标对象驱动部分的，就是计算机控制。

对此例中的控制来说，除了使它为某一目的进行必要的动作之外，还必须知道控制目标处于何种状态。在计算机控制中，此控制目标的状态，可通过传感器获得的信号来处理，这一信号经过计算机处理，变换为控制信号，这就是称为执行机构的控制机械动作的信号。如前所述，“传感器(sensor)”是相当于人的视觉、听觉、触觉等感官，是可以把光、压力、位移等物理量或化学量，变换为适合于以后处理的信号的器件，也称为检测器。“执行机构(actuator)”，是用电或液压等能量，使机器动作的装置。例如电动机、螺线管、油缸等。

II 计算机控制的特征

以下列举计算机控制的特征。

通过改变程序就能灵活地(柔性)适应控制对象。把控制装置作为真正的控制器来用，就必须判断装置所处的状态，根据传感器的输入信号即可判断，并通过计算机高精度地自动控制对象的运行。

计算机控制装置体积小、重量轻、并且驱动所需的电功率小，承受温度、灰尘等恶劣环境的能力较强，可靠性高。图 3 所示为计算机控制热处理炉的例子。利用来自传感器的信息控制加热电源的接通与切断，使炉内温度恒定。

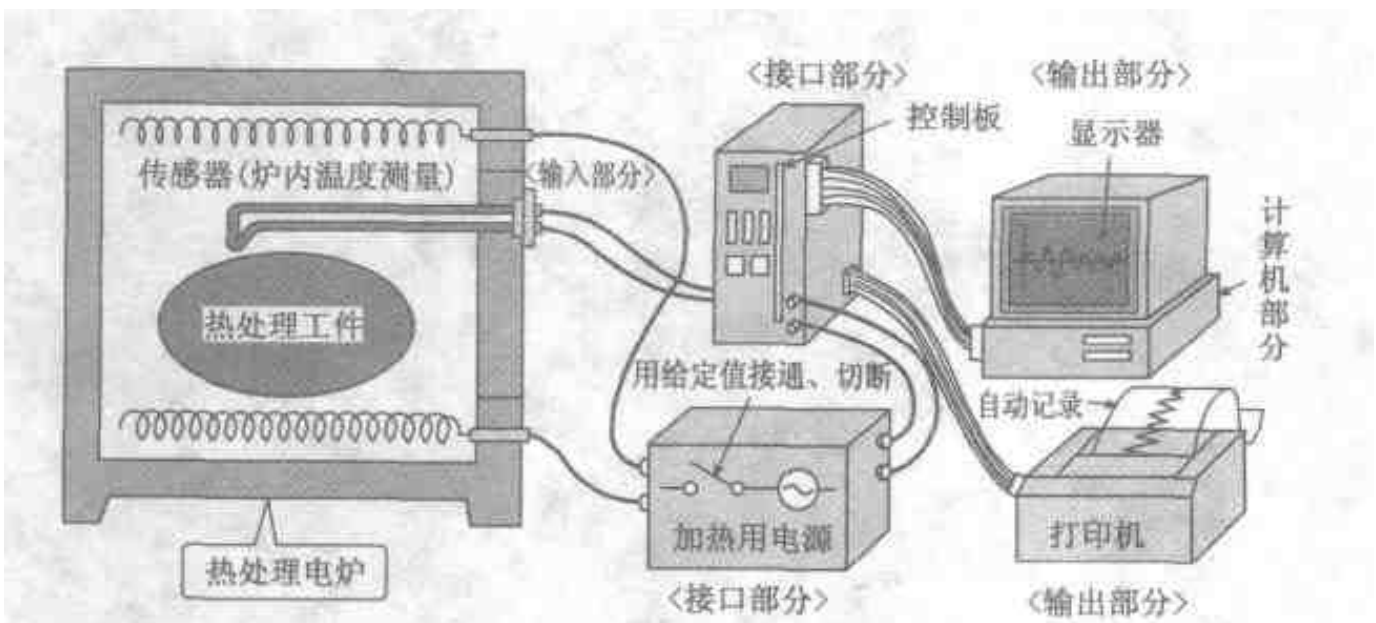


图 3 热处理炉的计算机控制

Ⅲ　计算机控制的组成

计算机控制如图 3 所示，由下列三大部分构成。

计算机

计算机由中央处理装置(CPU)、存储器与运算部件构成。

接　口

接口的作用是把炉内温度的高、低等信息，变换为计算机能处理的信号。

输入输出装置

输入装置是传感器，它通过温度检测元件(热电偶)检测炉内温度，并把信号送入接口，按照计算机发出的指令信号对加热电源进行控制。在图 3 中，打印机记录了炉内温度的变化，打印机是输出装置，当然显示器也是输出装置。

3 数字信号的定义

使用个人计算机的文字处理等软件时，不用知道计算机的内部结构或程序指令，也能对计算机进行操作。但是要用计算机进行控制，则必须理解“操作计算机的语言”。这里介绍一下关系到计算机动作的“1”、“0”数字信号。

Ⅰ 计算机处理的数值

我们生活中，用“十进制”表示数字。这里我们探讨一下表示电气开关状态时的情况。

如图 1 所示的开关，只有“ON－通”、“OFF＝断”两种状态。这一状态用数字表示则为“ON＝1”、“OFF＝0”。

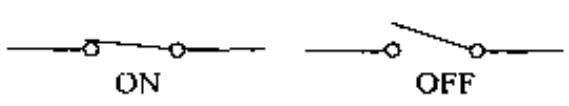


图 1　开关输入

像这样用“1”与“0”明确表示两种状态，所以也可从表示“有”、“无”的状态。

也就是说，计算机处理的数值，只有“0”与“1”两种，这种计算方法称为“二进制”。

Ⅱ 二值信号

可用“0”与“1”两种状态表示的信号称为二值信号。如图 2 所示，根据电灯的亮、熄或电压的大小等二值信号来表示状态的事物是很多的。计算机中所用的最小信号单位即 1 位(二进制的 1 位)，也是其中之一。把这样的“1”与“0”信号称为数字信号。

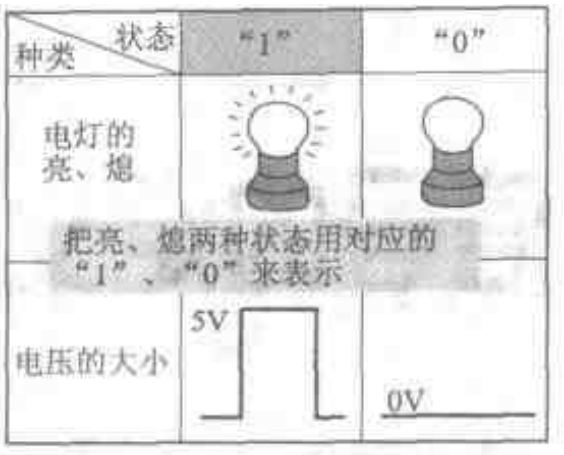


图 2　电灯的二值信号

Ⅲ　阈　值

在图 3 中，现有电压的大小为 3.5V，该如何判断是 1 还是 0 呢？这种场合下就得根据其状态的种类事先设定区分基准来判断"有"还是"无"，把这一作为基准的电平称为"阈值"(门限)。例如，如图 3 所示，以 5V 的一半 2.5V 作为阈值时所作的判断。这一阈值的大小是数字电路中极为重要的因素。

常用的 TTL 数字电路中，设置门限电压 1.35V 左右作为阈值。如图 4 所示，把约为 2.7V 以上的电平作为"1"，把约为 0.5V 以下的电平作为"0"。

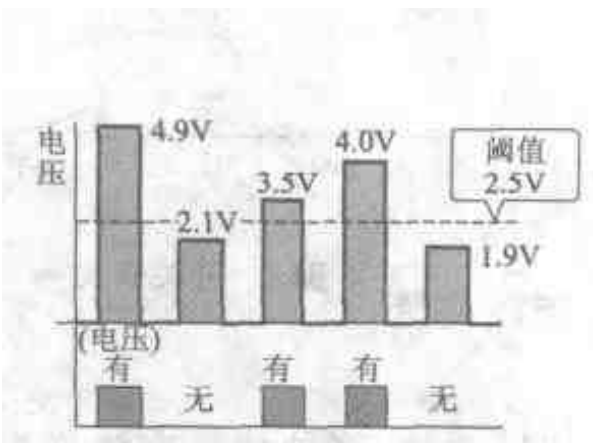


图 3　阈　值

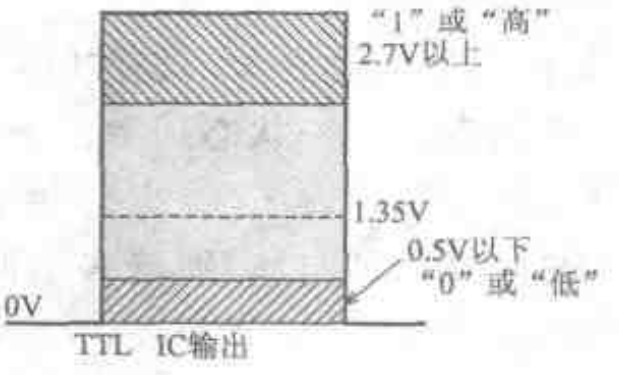


图 4　门限电平

有电压状态称为"高"电平，无电压状态称为"低"电平。

Ⅳ　二进制数、十进制数和十六进制数

十进制数到二进制数的变换

如图 5 所示，可通过求出十进制数中含有二进制数的位来求得。

另外，也可用图 6 所示的方法，将十进制数连续用 2 来相除，把所得到的余数以倒序排列，以此求得二进制数。

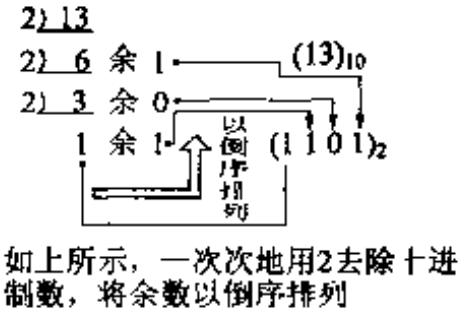
[例] 将十进制数13变换为二进制数

位权	2^3	2^2	2^1	2^0
	8	4	2	1
1或0	1	1	0	1

$(13)_{10}$ = $(1101)_2$

8+4+1=13

图 5 二进制



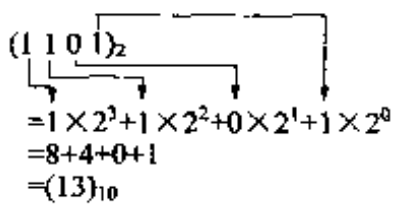
如上所示，一次次地用2去除十进制数，将余数以倒序排列

图 6 二进制数的求法

从二进制数到十进制数的变换

与从十进制数到二进制数的变换相同，可从给出的二进制数各位上的数(1或0)中，乘以“位权”所得之积的总和求之。

[例] 将二进制数$(1101)_2$变换为十进制数。



十六进制数

十六进制数(数的前面带有 &H 字符的)，从 0～9 与十进制数同样用法，10～15 则用 A～F 的英文字母表示，由于可用 4 位二进制数表示 1 位十六进制数，所以以 4 位来进行信息处理是很方便的。

表 1 是表示十进制数、二进制数与十六进制数的对照表。

表 1 十进制、二进制、十六进制对照表

十进数	二进数	十六进数
0	0 0 0 0	0
1	0 0 0 1	1
2	0 0 1 0	2
3	0 0 1 1	3
4	0 1 0 0	4
5	0 1 0 1	5
6	0 1 1 0	6
7	0 1 1 1	7
8	1 0 0 0	8
9	1 0 0 1	9
10	1 0 1 0	A
11	1 0 1 1	B
12	1 1 0 0	C
13	1 1 0 1	D
14	1 1 1 0	E
15	1 1 1 1	F
16	10 0 0 0	10
17	10 0 0 1	11

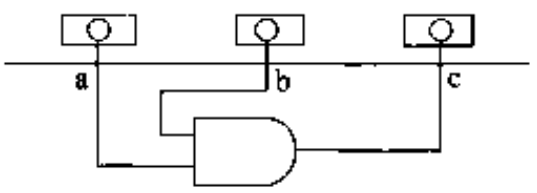
注：这里 A、B 不是通常意义上的英文字母。A 表示 9 加 1 所得的数，B 为再加上 1 所得的数。

例：&H1A4AD $=4\times16^4+10\times16^3+13\times16^0$

$=1024+160+13=1197$

4 逻辑电路实验装置的制作方法

例如把7408 (TTL-IC) 插入14脚插座中。



在a、b的插孔上，连接由开关SW输入的引线，把来自C孔的引线接到发光二极管 (LED) 的电路上去。这样，根据LED的状态就可以确定由开关 (SW) 状态所决定的逻辑关系，同时可以通过试验笔，检查各端子上的1与0的状态。

I 逻辑电路简介

要理解计算机控制，必须要有数字电路（逻辑电路）的基础知识，这里通过制作逻辑电路实验装置（以下称为 IC），一边用脑子来考虑，一边用眼睛与手，也就是身体来体会其中的逻辑电路。

IC 是用发光二极管（LED）的亮、熄来判断并理解逻辑关系的实验装置，首先它用基本逻辑电路 AND、OR、NOT 制成晶体管-晶体管逻辑集成电路 TTL IC（插入插座中），用引线连接各个不同的端子构成电路。

我想使用实际操作的实验工具 IC 也能更好地理解，完全懂得逻辑电路信号的作用。

*　　*　　*

要使用接口来制作电路，焊接作业是必不可少的，但在动手之前，奉劝大家务必先制作 IC 的逻辑电路实验装置，IC 的电路板参见图 1。

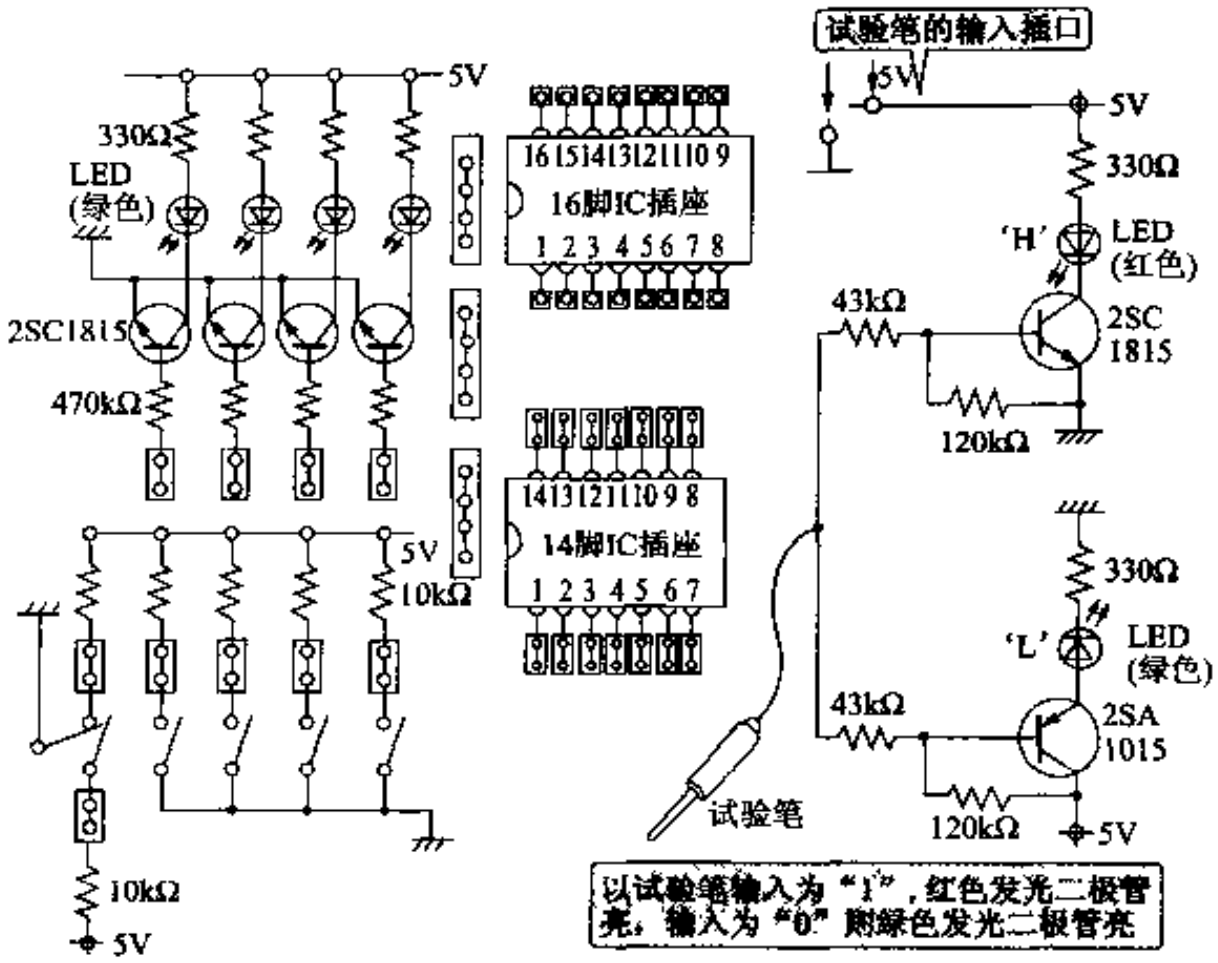


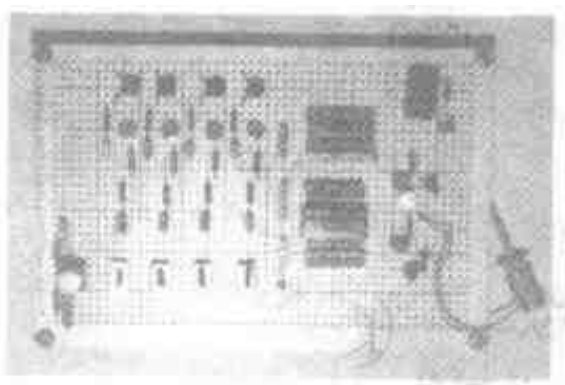
图 1　IC 电路图

Ⅱ　元器件表

基板	ICB-505	一块
晶体管	2SC　1815	5 只
	2SA　1015	1 只
电阻	330Ω	6 只
	470Ω	4 只
	120Ω	2 只
	43Ω	2 只
	10Ω	6 只
发光二极管	红	1 只
	绿	5 只
滑动开关		4 只
按钮开关		1 只
5V 电源连接器		1 只
跳接引线	长 12cm 左右	12 根
IC 印制板插座条(单列)		50 孔
IC 插座	14 脚	1 只
	16 脚	1 只
试验笔		1 支
IC	7408　带插座	1 只
IC	7432　带插座	1 只
IC	7404　带插座	1 只
IC	7400　带插座	1 只
IC	7402　带插座	1 只
IC	7486　带插座	1 只

Ⅲ　IC 的制作方法

元器件的总体布置见照片 1，将各部分分别排列即可。来自 5V 电源的正极线与地线的布线都利用基极四周的印制铜线。



照片 1　IC 电路布置

尽管电源用了易于购置的连接器，但使用装有四个 3 号电池的夹子在桌上试验也是很方便的。可根据自己的条件灵活选用。有可能的话，可把电源校核用的发光二极管接入电路中，便于在实验总是不顺利时判断“电源有没有接通”。

14 脚集成电路 IC 与 16 脚 IC 的 5V 电源与地线管脚均应分别各自配线。14 脚 IC 与 16 脚 IC 用的 1 号管脚均位于缺口的旁边，只有对准了缺口，实验用的 TTL-IC 才能插入。当然，焊接面与元器件安装面上的管脚号是相反的。

装滑动开关时，其方向应当为往上滑动时，所组成的电路使发光二极管点亮。

IC 印制电路板插座条中，有三处为单列 4 孔，这一部分是用于转接信号的，用于当一个信号有几个分支电路时。逻辑校验高(H)、低(L)电平，可用发光二极管的颜色来区别。

5 IC实验(1)
——AND电路的实验

将介绍的与、或、非这三种电路，都是最基本的逻辑电路，必须充分理解其作用。

Ⅰ AND(与)电路

图 1 的小灯泡控制电路中，是两个开关都接通“ON”的场合，小灯泡才亮起来。像图 1 这样有多个输入，只有它们全部都为“1”的场合，输出才为“1”的电路称为 AND 电路。

试一下身边的个人计算机键盘，例如仅当以下三个键同时按下时，才有意义(成为“1”)。如想结束窗口 Windows 操作时，是通过同时按下 Ctrl 键与 Alt 键和 F4 键(“1”＝功能有效，注：日式键盘)来实现的，这就是 3 输入 AND 电路的例子。

图 1　AND 电路

Ⅱ IC 的 AND 电路实验

将 IC 74LS08 插入 14 脚的插座中。74LS08 是由 4 个 2 输入 AND 电路构成的，通过 7 号管脚接地和 14 号管脚接电源来实现其功能(参见图 2)。

表 1 是 2 输入 AND 电路的输入与输出关系的真值表。真值表用于帮助理解“AND”输入组合的条件下，所对应的逻辑输出值，在考虑逻辑电路时，非常方便。

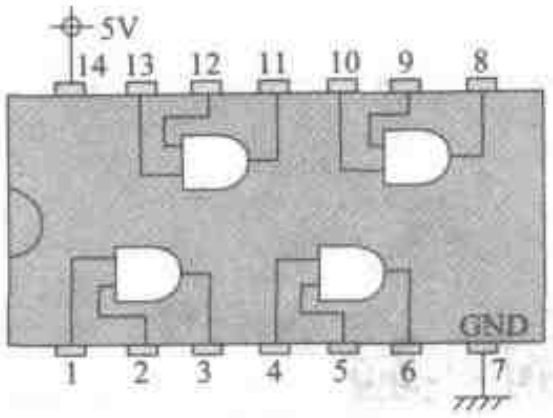


图 2　74LS08 的管脚排列

表 1　真值表

输入		输出
A	B	A+B→C
0	0	0
0	1	1
1	0	1
1	1	1



OR逻辑电路符号

把 5V 电源插入接插件上，电源指示灯发光二极管(LED)就亮。

如图 3 所示来连接引线。

A-IC 的 1 号引脚为输入脚；

B-IC 的 2 号引脚为输入脚；

C-IC 的 3 号引脚为输出脚，接 LED。

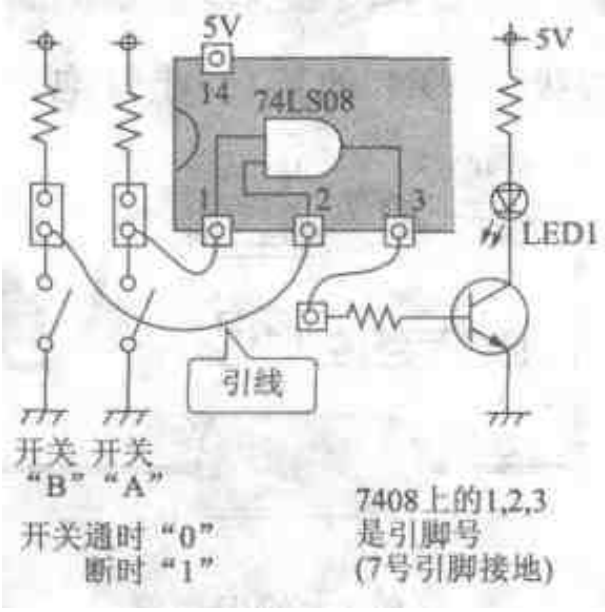


图 3　IC 的连接

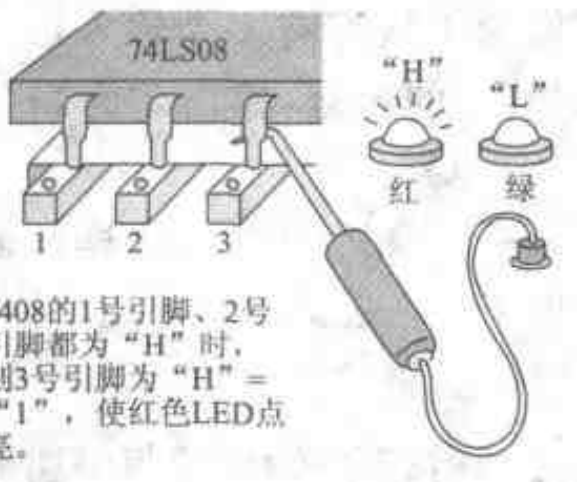


图 4　用实验笔的逻辑试验

现在如图 4 所示，当实验笔接触到 IC 的引脚时：

当开关接通(ON)时(接 0V GND* ="0")；

当开关断开(OFF)时(接 5V="1")。

逻辑实验的 LED 的显示分别为：

"1"时为红色 LED 亮；"0"时为绿色 LED 亮。

由此可知信号输入的状态。

IC 的 3 号引脚的工作状态，如真值表所示，当 1 号引脚与 2 号引脚的输入均为"1"时，3 号引脚的输出也为"1"。

其他的组合状态中，也可用 IC 确定其逻辑功能是否正常。

用引线将起到“与”电路作用的输出，接到 LED 电路上，如图 5 所示的晶体管导通，就可知道 LED1 应当点亮。

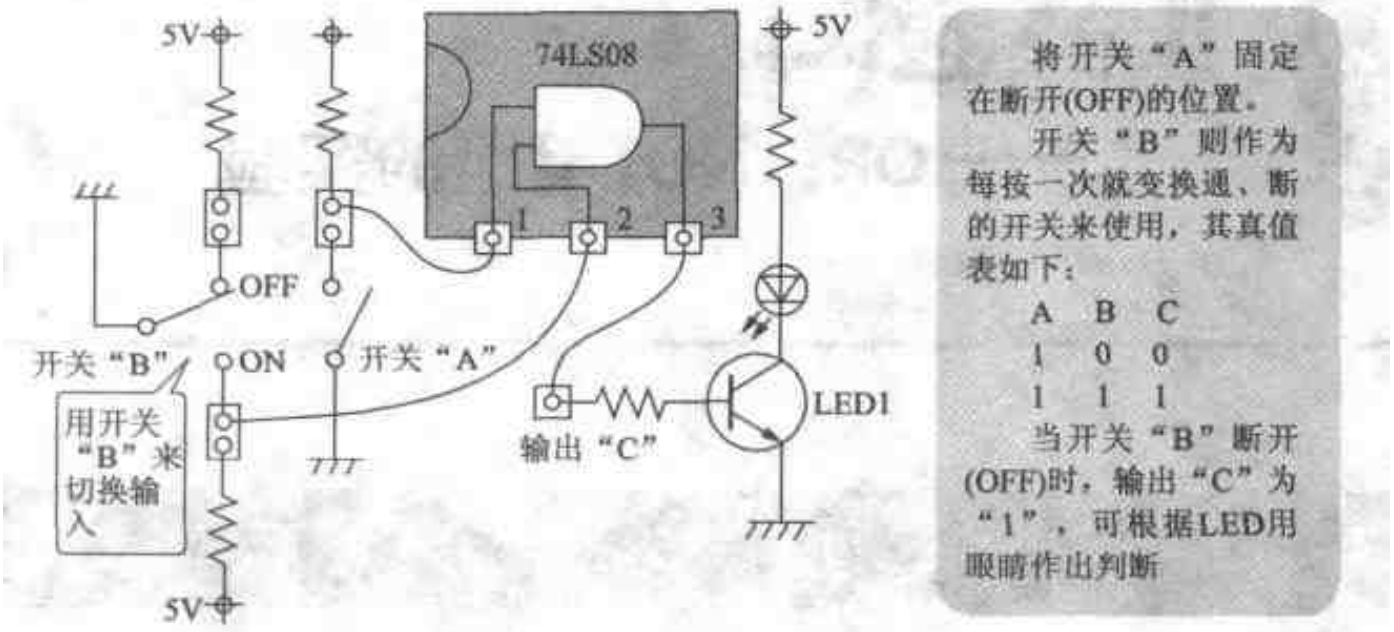


图 5　使用按钮开关的 AND 电路

6 IC实验(2)
——OR，NOT电路的实验

Ⅰ　OR(或)电路

如图 1 所示，小灯泡的工作电路中，只要两个开关中任意一个开关接通“ON”，小灯泡就点亮，当然两个开关都接通时，灯泡也亮。

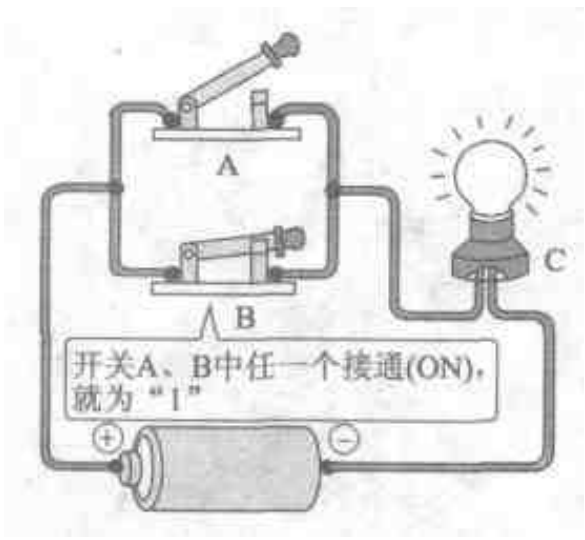


图 1　OR 电路

如图 1 所示，有多个输入中，其中任意一个为“1”时，输出就成为“1”的电路，称为 OR 电路。换句话说，“或逻辑”是“A 或 B 或……”中任意一个条件成立(这里的场合为“1”)就有输出的逻辑。若用我们所熟悉的串联与并联连接术语，则 OR 电路就与开关的并联连接相同，前面所介绍的 AND 电路则与串联连接相同。

举一个我们常遇到的 OR 电路的例子，在只有一名司机的公共汽车上，任何一个人按下显示“下车”的按钮，司机就知道有人需要下车。

Ⅱ　IC 的 OR 电路实验

将 IC 74LS32 插入插座中，74LS32 的引脚排列如图 2 所示。接好逻辑试验电路和连接引线，用试验笔就可进行如图 3 所示的 OR 电路逻辑的实验。

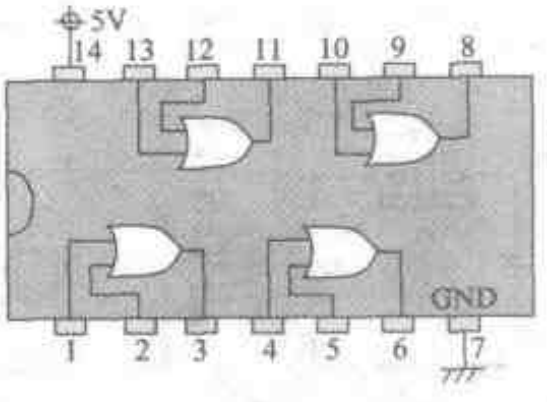


图 2 74LS32 的管脚排列

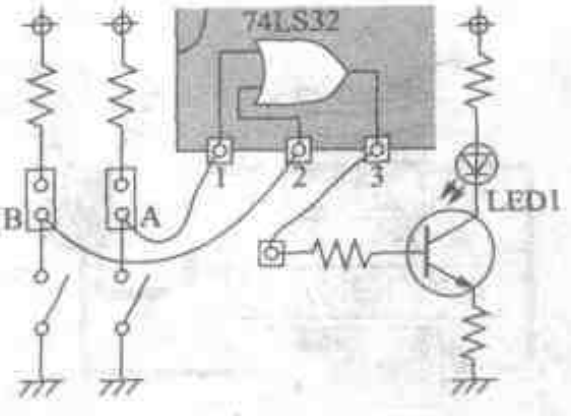


图 3 OR 电路的连接

表 1 是 OR 逻辑电路的真值表。

表 1 真值表

输入		输出
A	B	A+B→C
0	0	0
0	1	1
1	0	1
1	1	1



OR逻辑电路符号

Ⅲ NOT(非)电路

NOT 电路,相当于“否定”的意思,具有否定的功能。也就是输入为“1”时,输出为“0”,输入为“0”时,输出为“1”的逻辑电路。图 4 所示为小灯泡的 NOT 的电路。

电冰箱中的指示灯的点亮、熄灭功能,就相当于 NOT 电路。打开电冰箱的门时,门开关为“0”,冰箱的灯泡接通“ON”。

关闭冰箱的门时,门开关为“1”,冰箱的灯泡断开“OFF”。

如图 5 所示,驱动电动机的继电器电路之中,制动器接点也常用 NOT 电路。NOT 电路常用于使用 IC 的接口电路中。所考虑的理由是,使用反相信号的逻辑电路,非常适合用于程序或执行机构的控制电路中。另外,为了稳定输入信号,也要将两个 NOT 电路串联连接(反相的信号再反相,则得到原来的同相信号)。

NOT 电路也称为反相器。

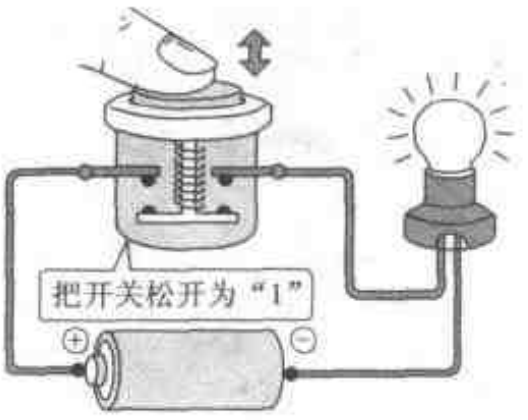


图 4　表示小灯泡的 NOT 电路

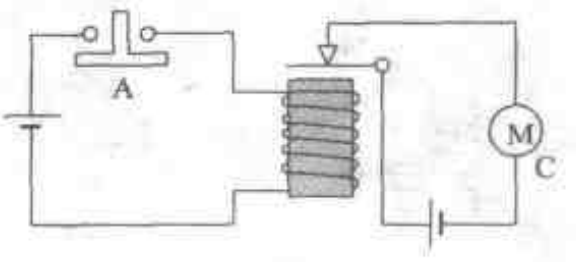


图 5　继电器电路的制动器接点

Ⅳ　IC 的 NOT 电路实验

把芯片 74LS04 插入 14 脚插座中，图 6 表示为 74LS04 管脚的排列，是由 6 个单一输入、单一输出的 NOT(非)电路构成的。NOT(非)电路的逻辑真值表如表 2 所示。

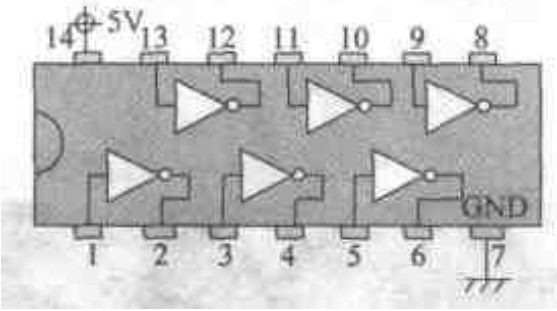
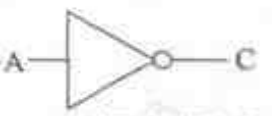


图 6　74LS04 的管脚排列

表 2　真值表

输入	输出
A	C
0	1
1	0



NOT逻辑电路符号

接好试验电路和引线，用试验笔就可以对图 7 所示的 NOT 电路进行试验。

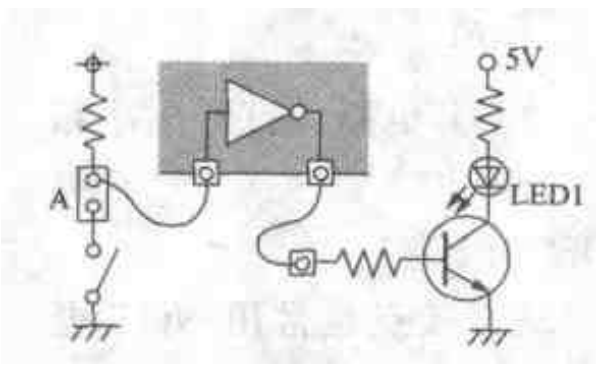


图 7　IC 的 NOT 电路试验

若将开关 A 接通(ON)与切断(OFF)，发光二极管就以其相反的状态来进行点亮与熄灭。

因为 AND、OR、NOT 电路是数字电路中最基本的电路，大家必须掌握其基本的使用方法。

7 IC实验(3)
——NAND，NOR电路的实验

Ⅰ　IC 组合逻辑电路

组合逻辑电路的定义是把几个基本逻辑电路(就是已学过的 AND、OR、NOT 电路)组合起来,成为单一目的工作状态的应用电路。常用的组合逻辑电路都是自成一体以一个 IC 来供货的。这里介绍的 NAND(与非)电路也是其中一例。

不过,把最简单的基本逻辑电路 IC 进行组合来构成电路,也是很重要的控制技术之一。我们通过"IC 实验",来确认其工作原理。

Ⅱ　NAND(与非)电路

如图 1 所示,NAND 电路是由 AND 电路和 NOT 电路连接而成的。

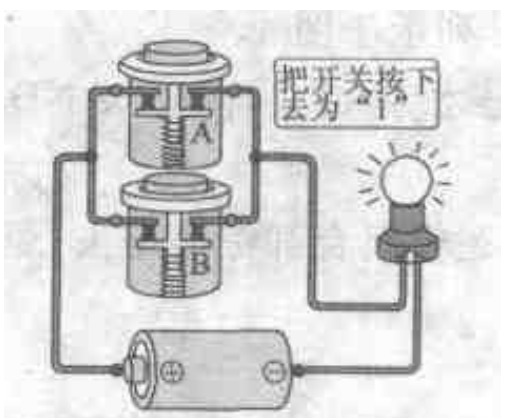


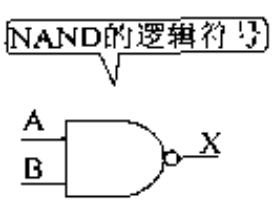
图 1　NAND 电路



图 2　NAND 电路的门"C"信号

表 1　NAND 真值表

输入		门	输出
A	B	C	X
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0



因 NAND 电路是由 AND 电路的 NOT(否定)组成的，所以其输出结果与 AND 电路完全相反。NAND 真值表见表 1 所示。

如图 2 所示，把这一信号流，作为中间有个门"C"的状态来考虑就好理解了。

Ⅲ　利用 IC 的 NAND 电路实验

如图 3 所示，将 IC74LS08(AND)插入 14 脚的插座中，将 74LS04(NOT)插入 16 脚的插座中，1 号管脚的位置一定要对准。因 IC74LS04 是 14 引脚，16 管脚的插座中还剩下 7 号脚与 8 号脚，要用跳线将引线连接起来，作为 14 管脚的插座来用。

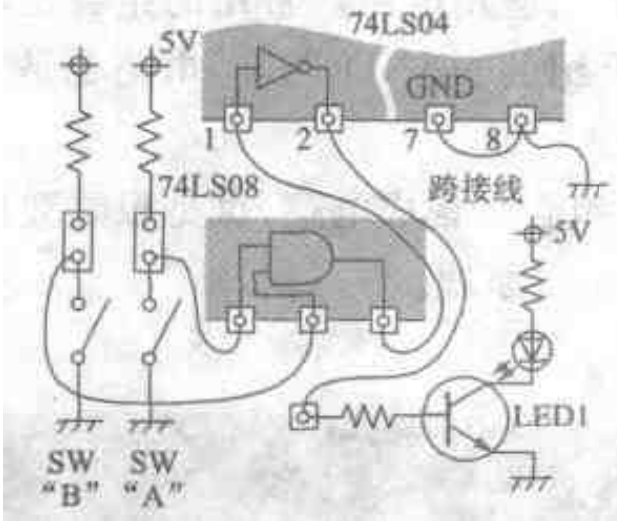


图 3　NAND 组合电路

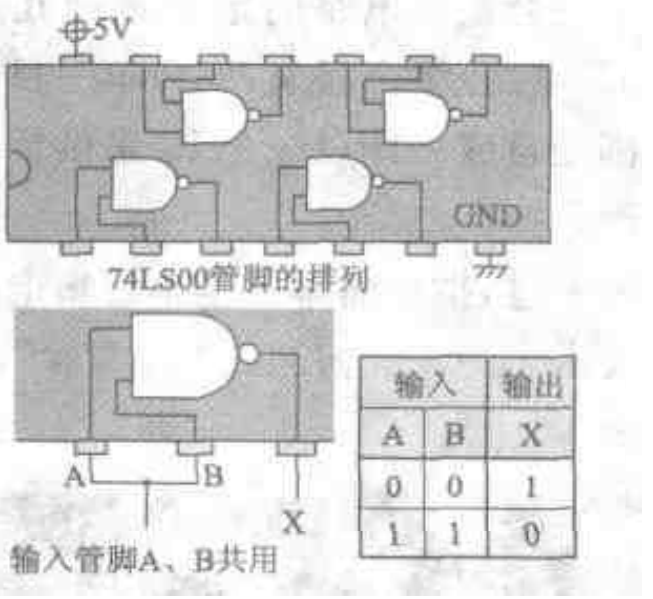


图 4　用 IC 逻辑板的 NOT 电路实验

双输入的 NAND 电路有 74LS00，其管脚排列示于图 1。

只有使这一 IC 的输入信号都为"1"时，输出才为"0"。即当 NAND 电路的输入端子都连通时，才成为非(NOT)电路。

像这样，通过考虑逻辑电路的各种组合，把逻辑组合都考虑进去，逻辑电路 IC 会有许多种使用方法。

Ⅳ　利用 IC 的 NOT 电路实验

将 IC 74LS00 插入 14 脚插座中。将 1 号管脚与 2 号管脚，如图 5 所示用跳

线连接起来作为共同的输入信号。

把来自开关的信号反转,就得到输出信号。这就是所说的 NOT 电路。

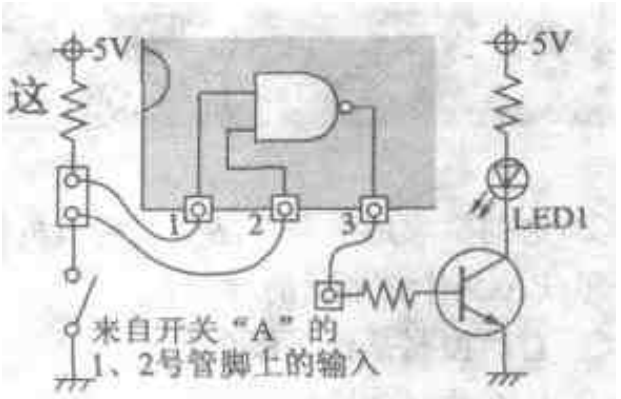


图 5 74LS00(自组)NOT 电路

Ⅴ NOR(或非)电路

如图 6 所示,NOR 电路是由 OR 电路上连接 NOT 电路组成的。NOR 电路的作用由图 7 的开关动作就可以明白,只要有一个开关为"1",输出就为"0"。换句话说,当输入全为"0"时,输出才为"1"。其真值表示于表 2。



图 6 NOR 电路的门"C"信号

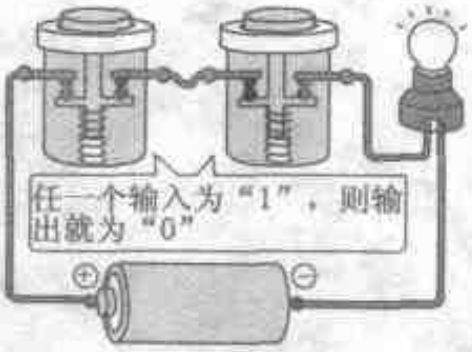
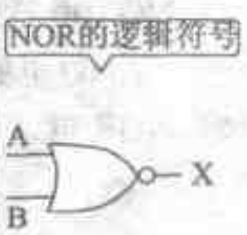


图 7 NOR 电路

表 2 NOR 真值表

输入		门	输出
A	B	C	X
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0



Ⅵ 利用 IC 的 NOR 电路实验

将 74LS32 插入 14 脚插座中,将 IC74LS04 插入 16 脚插座中,7 号脚与 8 号脚用跳接线连接。可用与 NAND 电路实验的相同顺序,确认 NOR 电路的工作状态。

Q 挑战题

① 请列举家用电器产品中使用微机控制的例子。
② FA 是什么术语的缩写?
③ 自动控制是什么概念?
④ 什么是双金属片?
⑤ 控制方式分哪两类?
⑥ 调速机是什么装置?
⑦ 列举顺序控制的例子?
⑧ 计算机控制中,用什么转换控制流程?
⑨ 列举计算机控制的特征。
⑩ 描述计算机的组成。
⑪ 什么是二进制?
⑫ 什么是十六进制?
⑬ 把十进制数"15"用二进制数与十六进制数分别表示。
⑭ 什么是阈值?
⑮ TTL 是什么?
⑯ AND 电路是什么样的电路?
⑰ OR 电路是什么样的电路?
⑱ NOT 电路是什么样的电路?
⑲ NAND 电路是什么样的电路?
⑳ NOR 电路是什么样的电路?

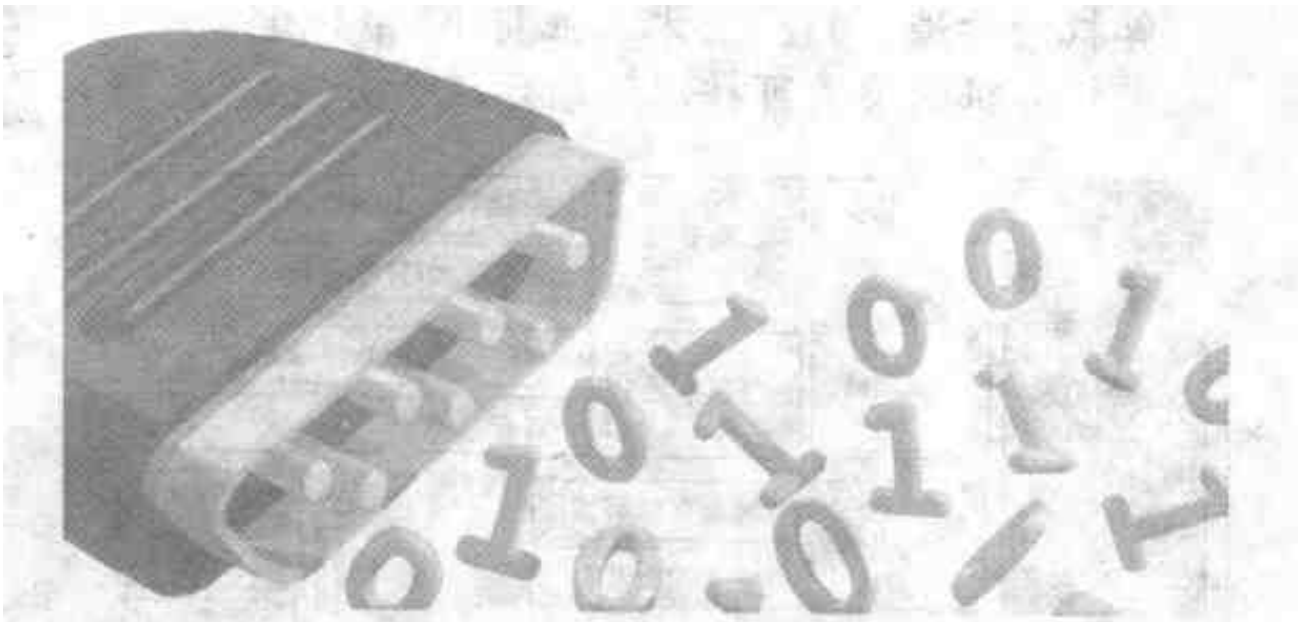


A 答　案

①电饭锅、空调机、电冰箱等。　②Factory Automation。　③参照第 1 节。　④两片膨胀系数不同的金属熔接在一起的制品,热变形。　⑤顺序控制、反馈控制。　⑥参见第 1 节。　⑦交通信号控制器、洗衣机。　⑧变换程序。　⑨参见第 2 节。　⑩由计算机、接口与输入输出部分构成。　⑪用 0 与 1 表示数字大小。　⑫参见第 3 节。　⑬二进制数 1111,十六进制数 F。　⑭即成为状态基准值的电平。　⑮晶体管-晶体管逻辑。　⑯两个以上的输入均为 1 时才产生输出 1,对于其他输入,输出均为 0。　⑰两个以上的输入中任意一个都为 0 时输出才为 0,其他情况均输出 1。　⑱将输入信号反相作为输出。　⑲若任意一个输入为 0,输出为 1,若全部输入为 1,则输出为 0。　⑳仅全部输入为 0 时,输出为 1。

第5章

计算机与接口



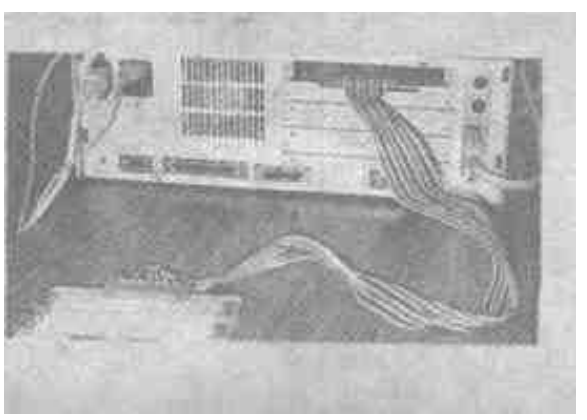
计算机的基础理论与控制

学习计算机控制，一定要学好计算机的结构和与接口的有关知识。最早的计算机是1941年美国的阿塔那索夫的自动二进制计算机，当时的重量是以吨为单位计算的。现在的计算机不但重量很轻，体积也小，相反地性能却有明显的提高。

现在计算机用于文字处理与游戏等领域，从大人到小孩均享受到其益处。同时，手表、电视机、电饭锅、血压计、汽车等我们身边的机械都用到了计算机。在日常生活中，我们不知不觉地在使用着计算机。

计算机具有性能超常、体积很小、价格低廉的特点，已成为控制中不可缺少的装置。在机械设计中，如果控制装置不引入计算机，不仅不能满足用户的要求，而且落后于时代了。

要在计算机上接入控制用的机器，就必须使用接口。接口就是界面的意思，是用于把连接器件的信号调整为能顺利地传送的装置。



在计算机中，连接的接口是接在机箱内部扩展槽上的。数据总线、地址总线、控制总线等相应的端子均必须连接好。

这里用8255接口进行信号传送控制。8255是一个8位并行的输入输出接口。

1 计算机的构造

在学习计算机之前，先简单介绍一下计算机的构造。JIS(日本工业标准)中规定，构成计算机的要素可按以下所示进行分类。





I 处理器(处理装置)

处理器相当于人的大脑，由处理单元与内存组成(参见图 1)。处理器也称

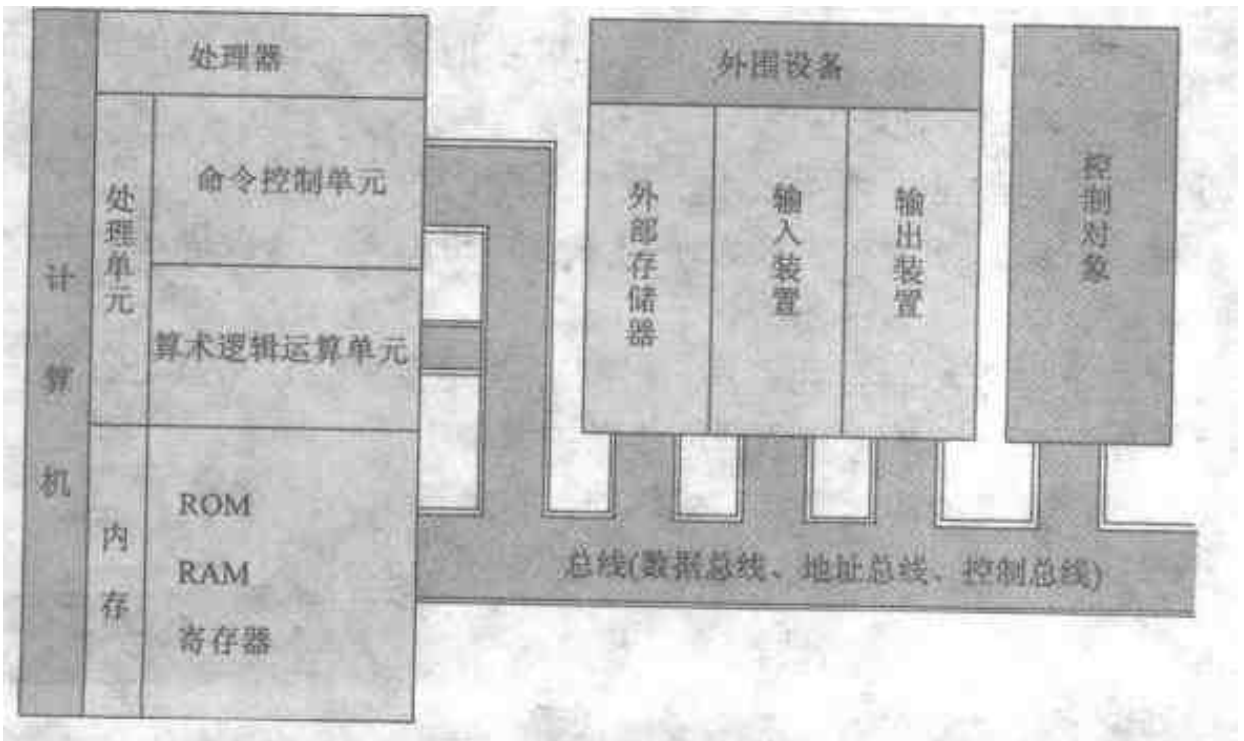


图 1　计算机的构造

中央处理器(CPU),把集成化的中央处理器称为微处理器(MPU)。处理单元由控制单元与运算单元组成,通过总线(BUS)来连接。

① 控制单元

顾名思义,就是控制计算机的单元,它的作用是解释进入内存的程序,根据解释向运算单元或其他单元发出相应的命令,完成计算机所要完成的工作。

② 运算单元

运算单元是根据控制单元的命令进行算术(+、-、×、÷)或逻辑(与、或等)的运算,或作出判断等等,是计算机主要的工作部件。

③ 内存

内存相当人脑的记忆部分,是不通过输入输出通道的存储器,由能存储大量数据的 ROM(read-only memory,只读存储器)或 RAM(readom-access memory,读写存储器),及以一个字(字节)为单位或以一位为单位的临时存储的寄存器等组成。

④ 总线(BUS)

总线是多个装置之间相互连接用的信号线。它不仅连接处理单元内部,也与外围设备等连接,进行数据传送。

数据总线:是传送数据的信号线。数据包括程序、处理数据、输入输出数据等。

地址总线:是传送地址数据的信号线。地址包括存储器地址、集成电路芯片地址等。

控制总线:是传送控制数据的信号线,传送各装置工作用的操作信号等。

Ⅱ 外围设备

是指实际工作中除处理单元外计算机所需要的装置,包括输入输出装置或外部存储器等。

① 输入输出装置

输入输出装置是计算机与人或机器交换数据的装置,包括:

- 输入数据的键盘或鼠标等输入装置。
- 把数据以图像形式显示的显示器(CRT:阴极射线管),或打印机等输出装置。
- 其他与外部交换数据的各种接口等。

② 外部存储器(辅助存储器)

因计算机要处理大量的数据，必须有硬盘与软盘等外部存储器储存数据。

③ 控制对象

计算机的控制对象，一般通过接口连接在处理单元的各种总线上。

PC9801 的扩展槽

用个人计算机进行控制的场合，可使用扩展槽的各种总线插座。PC 9801 系列的个人计算机的扩展槽中，里外合计共有 100 根线的端子插座，其中有数据总线 16 根、地址总线 24 根，以及控制总线、电源线等。根据计算机型号的不同，总线的功能也有所区别，应当根据使用说明书确认其用法(图 1)。

练习题

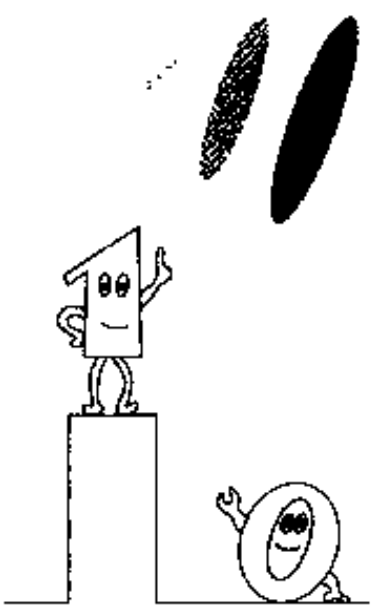
1. 作为计算机的重要部件，即 CPU 部件，是何意？
2. 计算机内外传送数据的线叫什么？
3. 外部存储器有什么必要？
4. 用个人计算机控制的场合，连接在计算机的哪个部分？

解答

1. 处理器(处理装置)
2. 是总线(BUS)，有数据总线、地址总线、控制总线等。
3. 为储存大量的数据。
4. 把输入输出信号连接在个人计算机内部的扩展槽中。

2 计算机的信号(1)

通过计算机各种装置内部与外部的信号有高电平与低电平两种电信号；高电平表示数值1，低电平表示数值0，以这种1与0表示的信号称为数字信号。用这种信号工作的计算机称为数字计算机。现在的计算机，普遍都是指数字计算机。数据总线、地址总线或控制总线等所有总线，都传送数字信号。



Ⅰ 数字信号

信号有数字信号与模拟信号之分，数字信号有“可数”的意思，即意味着可用整数等非连续数值表示的信号。

数字计算机中的数字信号只能用1与0的数值表示，1与0的数字信号可表示出从输入输出信号到程序的所有数据。计算机只处理“1”、“0”这两个基本信号。

Ⅱ 数据总线

用于传送处理数据的总线是数据总线。1根数据总线只传送1、0两种数据。要处理大量的数据，就要增加同时处理信号的总线数量。以4根信号总线为基础，有8根、16根、32根、64根等各种数据总线的计算机。总线根数越多，可同时处理的数据就越多。16根数据总线的计算机，称为16位计算机，32位计算机，则有32根数据总线进行数据处理。

PC 9801 扩展槽

扩展槽的数据总线管脚有16位机型的，也有32位机型的。PC 9801是16只(16位)管脚的，在32位总线机型的场合，可使用低端16位的数据总线。如要增加外部存储器等有必要用到32位的场合，因个人计算机内部有32位的数据总线，所以也可以与该总线管脚连接。

Ⅲ　地址总线

这是传送地址数据的总线，这些地址是存储器的地址或连接各种装置(大规模集成电路LSI)等的地址。计算机系统越大，使用存储器或装置种类越多，地址数也越多。地址总线多时，可使用从16根到32根的总线。

PC 9801 扩展槽

它有24根地址总线管脚，可传送24位地址数据。输入输出(I/O)用的地址为低端16位的也有效。个人计算机内部使用的地址越多，用户所能用的地址就受到限制，应当查阅操作手册或向制造厂查询清楚。

Ⅳ　控制总线

控制总线是信号传送线，用于传送控制计算机内外各种装置所必须的操作信号或装置的状态等信号。

控制总线每根均具有各自的作用，如复位或中断等。控制总线不一定全部使用，仅用所需要的控制信号。必须注意，控制信号1所具有的意义(在正逻辑中，高有效，表示高电平，在负逻辑中，则为低有效，表示低电平。)

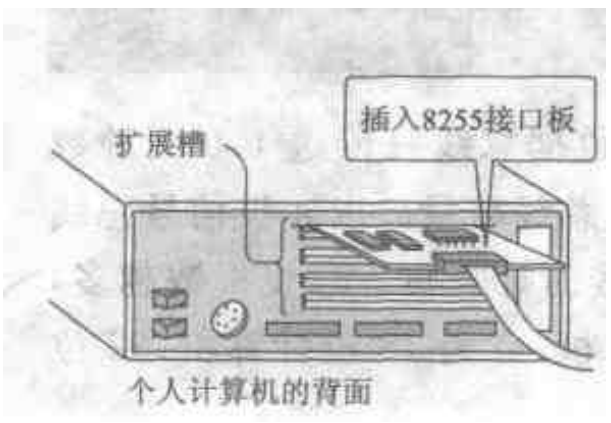


图1　扩展槽

PC 9801 的扩展槽

PC 9801扩展槽中有38只管脚的控制总线，应根据需要连接(参见图1与表1)。要想正确地使用信号，应当向制造厂索取有关硬件的资料。

表4 PC 9801 扩展槽信号一览表(例)

端子序号	信号名称	方向	功能	端子序号	信号名称	方向	功能
A1	GND			B1	GND		
A2	V1			B2	V1		
A3	V2			B3	V2		
A4	AB00	IO	地址总线	B4	DB001	IO	数据总线
A5	AB01	〃	〃	B5	DB011	〃	〃
A6	AB02	〃	〃	B6	DB021	〃	〃
A7	AB031	〃	〃	B7	DB031	〃	〃
A8	AB041	〃	〃	B8	DB041	〃	〃
A9	AB051	〃	〃	B9	DB051	〃	〃
A10	AB061	〃	〃	B10	DB061	〃	〃
A11	GND			B11	GND		
A12	AB071	IO	地址总线	B12	DB071	IO	数据总线
A13	AB081	〃	〃	B13	DB081	〃	〃
A14	AB091	〃	〃	B14	DB091	〃	〃
A15	AB101	〃	〃	B15	DB10	〃	〃
A16	AB111	〃	〃	B16	DB111	〃	〃
A17	AB121	〃	〃	B17	DB121	〃	〃
A18	AB131	〃	〃	B18	DB131	〃	〃
A19	AB141	〃	〃	B19	DB141	〃	〃
A20	AB151	〃	〃	B20	DB151	〃	〃
A21	GND			B21	GND		
A22	AB161	IO	地址总线	B22	+12V		
A23	AB171	〃	〃	B23	+12V		
A24	AB181	〃	〃	B24	IR31	I	INT0
A25	AB191	〃	〃	B25	IR51	I	INT1
A26	AB201	〃	〃	B26	IR6	I	INT2
A27	AB211	〃	〃	B27	IR91	I	INT3(硬盘)
A28	AB221	〃	〃	B28	IR101	I	INT4(640 软盘)
A29	AB231	〃	〃	B29	IR 121	I	INT5
A30	INT0	O		B30	IR 131	I	INT6(鼠标)
A31	GND			B31	GND		
A32	IOCHK0	I	外部 NMI	B32	-12V		
A33	IOR0	IO	命令	B33	-12V		
A34	IOW0	〃	〃	B34	RESET0	O	复位
A35	MRC0	〃	〃	B35	DACK 00	O	5"硬盘
A36	MWC 0	〃	〃	B36	DACK 30	O	AUX
A37	INTA 0	IO		B37	DRQ 00	I	5"硬盘
A38	NOWAIT 0	I		B38	DRQ 30	I	AUX
A39	SALEI	IO		B39	WORD 0	I	
A40	MACS 0	I		B40	EXHRQ 10	I	
A41	GND			B41	GND		
A42	CPUENB 10	IO		B42	EXHLA 10	O	
A43	RFSH 0	O	刷新信号	B43	DMATC 0	O	过程结束
A44	BHE 0	IO		B44	NMI 2	IO	
A45	IORDY 1	I	就绪信号	B45	MWF 0	IO	
A46	SCKL 1	O	9.8304/7.9872MHz	B46	EXHLA 20	O	
A47	S 18 CLK 1	O	307.2kHz	B47	EXHRQ 20	I	
A48	POWER 0	O	电源确认信号	B48	SBUSRQ 1	O	
A49	+5V			B49	+5V		
A50	+5V			B50	-5V		

练习题

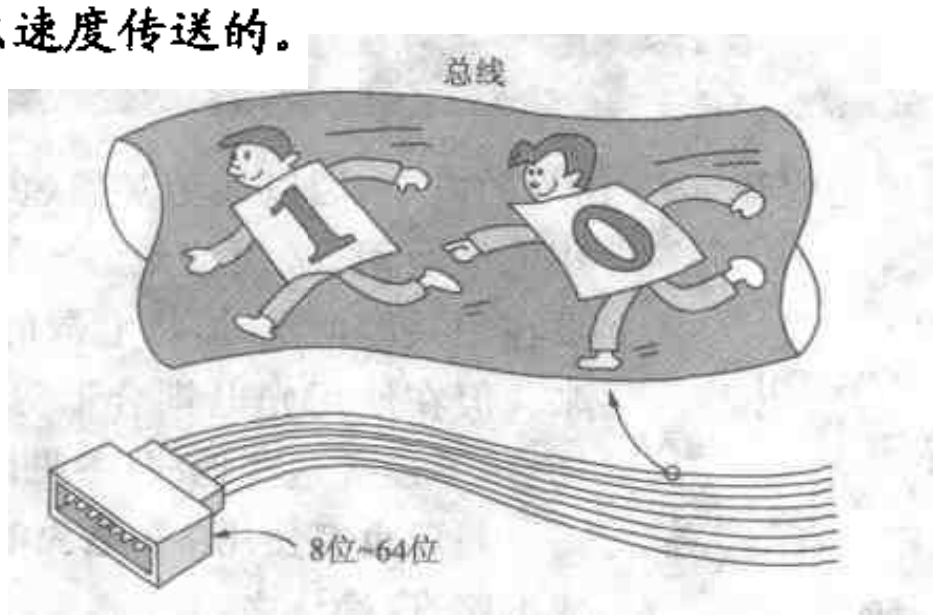
1. 计算机处理的信号是模拟信号还是数字信号?
2. 计算机处理的信号是什么样的电信号?
3. 32 位的计算机,有多少根数据总线?
4. PC 9801 的扩展槽有多根数据总线?

解答

1. 数字信号。
2. 直流 5V 作为电源,信号电压"1"为 5V,"0"为 0V。
3. 32 根。
4. 16 根。

3 计算机的信号(2)

在计算机内外传送的信号，实际上是电信号，探讨一下这是什么样的信号，以什么速度传送的。



I 信号的电气特征

计算机中，流经总线的电信号是电压为 5V 的直流电流，计算机处理的数字信号电压均为：高电平信号 5V；低电平信号 0V。

各电压值的许用值，均符合 TTL IC(晶体管-晶体管逻辑集成电路)的电平，但根据所用的集成电路 IC 的不同，也有用不同电压值的。其标准值如下(详细数据可参见有关个人计算机的硬件资料)：

电平许用值

输入信号

高电平信号＝1　　2.0～5V

低电平信号＝0　　0～0.8V

输出信号

高电平信号＝1　　2.4～5V

低电平信号＝0　　0～0.4V

信号电流的最大值

输入信号	1信号	20μA
	0信号	400μA
输出信号	1信号	−0.4mA
	0信号	8mA

（负的电流表示从IC管脚流出来的电流）

应当记住，计算机内外所流经的信号电流是很微弱的。

Ⅱ　电　源

信号用的电源是直流5V。信号的电平，因以地为基准（负电流），与外部电压电平通用，必须接地。不同电源的场合，也要在用共同信号时，将电源的地线互相连接起来。但在使用光电耦合器等隔离信号连接的场合，不连接这些地线（参见图1）。

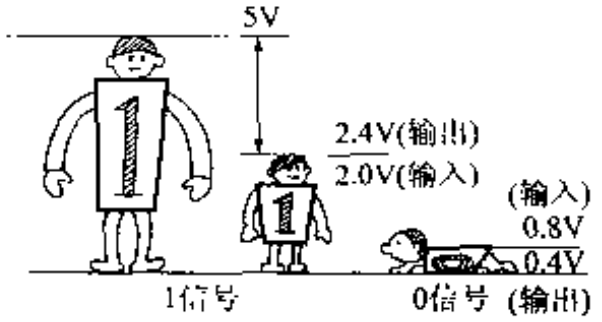


图1　信号的电平

最近也有使用3V电源的微型计算机或集成电路，应确认之。

Ⅲ　处理速度

计算机的处理时间，是以计算机所用的时钟脉冲周期（机器周期）为基准的。例如，时钟频率为10MHz的场合，周期为1/10 000 000（秒）=0.1μs，把它称为1拍（1个机器周期）。处理时间是它乘以倍数所得的值。根据语言或命令的种类不同，处理时间约从数微秒到数百微秒，汇编语言最快，而BASIC语言是边逐行翻译机器原码边执行的，所以很费时间。

因计算机与外围设备的处理时间不同，所以读取信号时，在延时时间上配合不好。为此采用以下方法：使用计算机上的时钟（系统时钟）作为外围设备控制信号的延时时间，来收发数据信号，或采用保持输出状态，使它符合接收方的延时要求。

用PC 9801扩展槽

• 若与外围设备连接线的长度超过某一长度（约500m）时，各总线的信号就会产生噪声或衰减，影响可靠性，必须将信号放大。

• 时钟频率随机型而不同，从 8MHz 到 200MHz 的都有，并且还有两种时钟的机型，应注意处理(参见图 2)。

• 电流管脚用了接地管脚 8 只，5V 管脚 4 只，+12V 管脚 2 只，-12V 管脚 2 只。

电流容量可供接口等作信号电源用，每一槽的 5V 电源约 0.8A，12V 电源约 0.06A，8 个接地管脚相互连接，是公用的(要确认)。

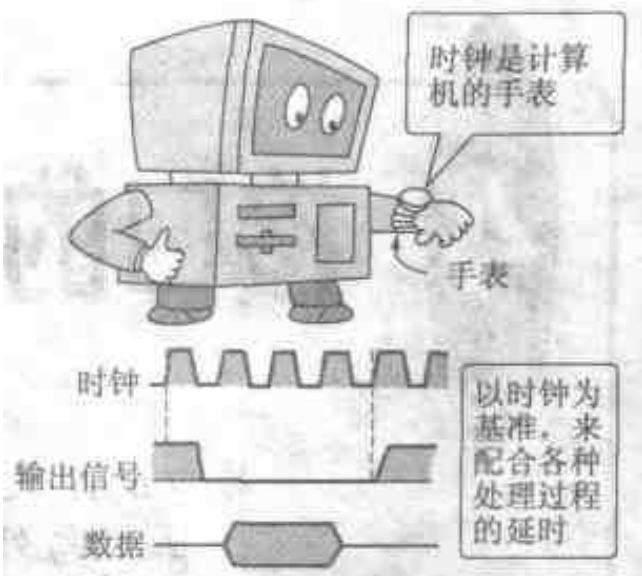


图 2 计算机的时钟

练习题

1. 信号电压的许用值，在输入信号与输出信号之间有区别吗？为什么？
2. 信号电流大约是多少？
3. 计算机的处理时间以什么为基准？
4. PC 9801 的扩展槽引出的电源容量，标准大概是多少？

解答

1. 信号电压的许用值，在输入信号时取宽一些，易于接收。
2. 信号电流为 mA 级到 μA 级。
3. 以计算机所带的时钟为处理时间的基准。
4. 每一个槽的电流容量为 5V 电源约 0.8A，12V 电源为 0.06A，有时根据机型而有所不同，需要确认。

4 接口的定义

接口的定义如下：

计算机与外围设备连接时，用于稳定与调整二者之间信号所必须的装置称为接口。这里介绍有关接口的功能。



1 电气条件的调整

计算机的信号虽然为直流 5V，但所用的各种外围设备的电源有交流 100V 或 200V，直流 12V 或 24V 等，必须对电源的种类与电压或电流等进行调整（参见图 1）。

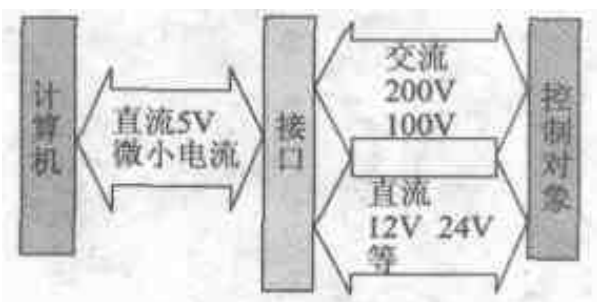


图 1　电气条件的调整

电压的放大或变换

要用计算机输出的信号使外围设备工作，必须使用各种继电器、晶体管或三端双向可控硅等，并用信号电压来驱动晶体管或继电器，以便对不同电压的直流或交流进行变换，还要用放大器等来放大传感器之类微小电压（mV 级）的变化。

电流放大

从计算机输出信号的电流很小，所以要向外部传送信号时，需要用集成电路（74 LS 245 等）或上拉电路等来放大电流，在驱动直流电动机或直流电磁继

电器等时，要用晶体管等来放大。

Ⅱ 信号变换

A/D(模/数)转换

计算机只能处理一连串的数字信号，而外围设备则大多数是处理模拟信号的，必须对相应的信号进行转换，这就是将模拟信号转换为数字信号。要将传感器等的模拟信号输入计算机时，必须进行这种转换。

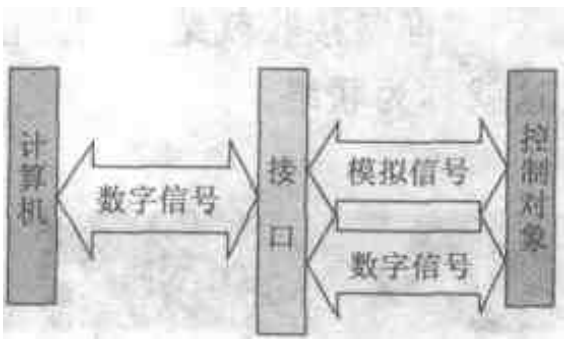


图 2 信号的变换

D/A(数/模)转换

这是将数字信号转换为模拟信号。在一些要连续地控制电动机转速的场合，就要将计算机输出的断续的数字信号转换为外围设备用的连续的模拟信号。

信号波形整形

要将信号波形进行必要的整形(参见图 3)。

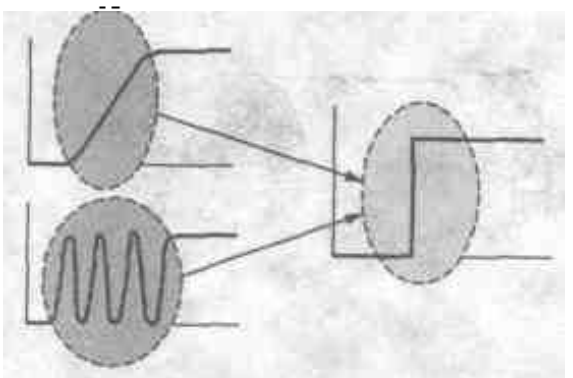


图 3 信号波形的整形

• 波形整形是将信号波形整形为高、低(1、0)分明的信号电压波形。若计算机或数字集成电路的输入信号不在信号电压的允许电压范围内，则不能判断高、低电平。因此要将连续的电压变化波形，整形为断续的波形。并且，抖动的波形是高低反复变化的，所以在耦合不良的情况下，要去除抖动波形。

可用积分电路或施密特触发电路等进行波形整形。

• 想要将瞬时的输入信号变换为有一定宽度的输入波形，或要将连续输入的波形变换为短时的输入波形等，也要用到微分与积分电路。

另外，要将计算机或输入设备来的瞬时信号，暂时保存在寄存器中，也要用到将波形拉宽的方法。

信号交接处理

在信号交接过程中，若发送端与接收端的延迟时间不一致，信号传送就不能进行，为此要使用控制信号，对诸如下列的对话进行信号传送：

“准备接收就绪”　“此信号有效”“正在接收信号”　“信号接收完毕”

这就称为联络。

防止干扰

因为计算机的信号微弱，易受到内外噪声的干扰，所以必须特别注意防止干扰。其对策有防止噪声源的产生，吸收易窜入的电源部分的噪声与防止噪声进入计算机等。

防止噪声干扰的方法有，在电源正极与电源负极(地)之间接入吸收噪声的电容器，或将信号线用光电耦合器等进行电气上的隔离等。但必须注意光电耦合器的信号流方向必须是单方向的。

练习题

1. 接口的主要作用是什么？
2. 为什么必须进行 A/D 转换或 D/A 转换？
3. 为什么必须进行信号波形整形？
4. 什么是信号交接处理？

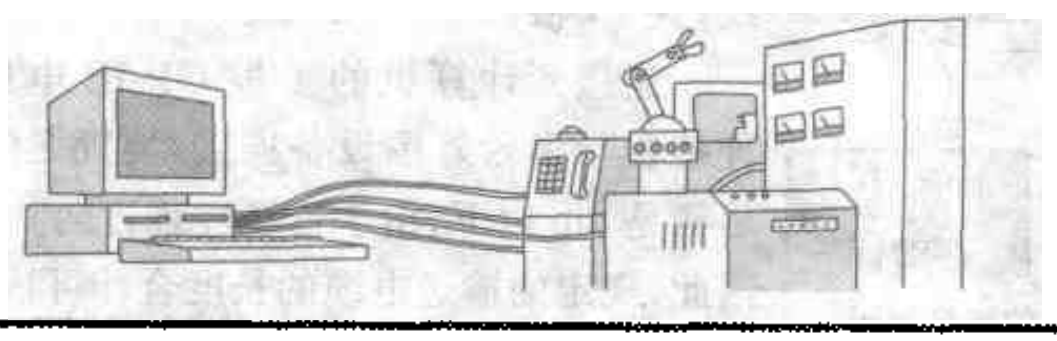


解答

1. 有电压或电流的放大或变换，A/D 或 D/A 等信号变换等。
2. 因计算机只能处理数字信号。
3. 为将计算机等处理的信号整形成为允许电压内的波形。
4. 在传送数据时，为了延时时间的一致，而用控制信号进行对话。

5 数据传输标准与通用接口

计算机的数据传输方法，为使其具有通用性，在标准化方面做了很多工作。现在简单说明一些有代表性的标准。



1 RS-232C 标准

RS-232C 标准是美国电子工业协会(EIA)的串行(串联)数据传输用的数据通信标准，日本工业标准 JIS C6361 中大致采用了同样的标准。几乎所有的个人计算机均以 RS-232C 接口作为标准设置，因此是有代表性的通信标准。

在计算机与控制对象的连接距离长的控制场合，要用 RS-232C(参见图 1)。RS-232C 接口将来自计算机的 8 位并行数据转换为 8 位串行数据，用通信电缆来传输。接收端将收到 8 位串行数据，用 RS-232C 转换为 8 位并行数据输入计算机中。

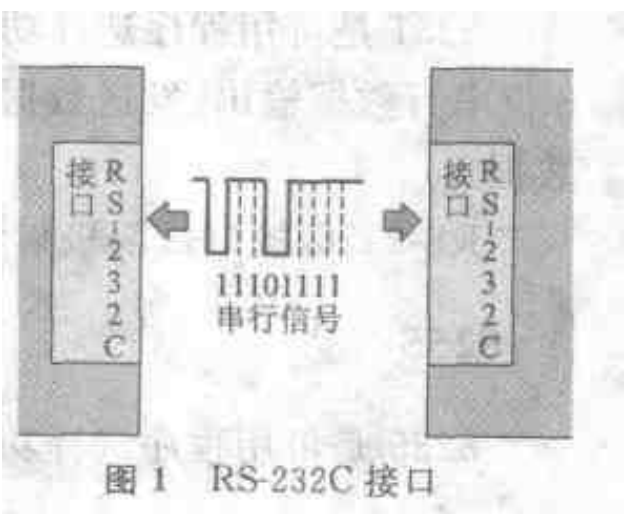


图 1 RS-232C 接口

大多使用 8251 集成电路来进行串并行转换。传输距离最大规定为 15m，实际上 15m 以上的距离也能传输，但距离越长可靠性越差。更长距离的数据传输则应使用 RS 422 接口或 RS 485 接口。

II　GP-IB

GP-IB 是计算机与检测装置或检测装置相互之间传输数据用的 8 位并行接口。GP-IB 是美国 HP 公司取的名称。在美国电气和电子工程师协会(IEEE)标准所规定的名称为 IEEE-488 标准总线。

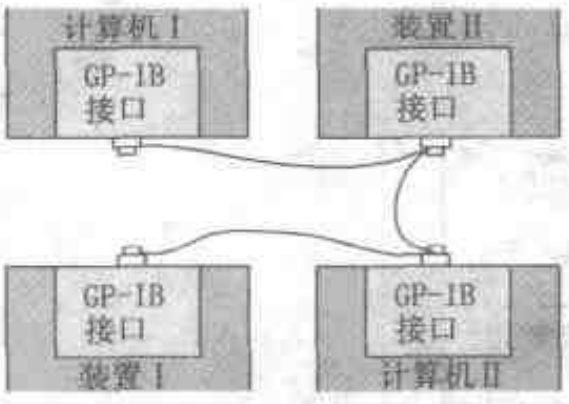


图 2　GP-IB 的连接例子

信号用 TTL 电平，采用负逻辑，信号线有 8 根数据总线、3 根联络总线、5 根管理总线、8 根地线，共 24 根(参见图 2)。

连接到计算机的 1 根 GP-IB 电缆，可经过各种装置与多台外围设备连接，装置之间的传输距离尽管是 4m 以下，由于可经过装置向其他装置传输，因此，规定的输送电缆的长度合计可达 20m，并且连接装置的规定台数包括计算机最多可达 15 台。

III　主要外围接口集成电路(IC)

下面介绍制作计算机接口时常用的 IC。

8251

8251 是可用程序选择功能的串行数据通信 IC。可将 8 位并行数据转换为 8 位串行数据输出、发送数据，同时可将接收到的 8 位串行数据转换为 8 位并行数据。

RS-232C 接口或阴极射线管等的接口等都要用到 8251 IC。

8255

8255 是可用程序选择功能的并行输入输出接口 IC，详见第 6 章的说明。

8259

8259 是中断控制用的 IC。有 8 只中断请求输入管脚，根据用程序规定的优先级别，向计算机提出中断请求。中断请求的输入端子优先级别，中断屏蔽(禁止中断)与中断矢量地址的指定均通过程序进行。中断优先级别与中断屏蔽均可由程序进行实时变更。

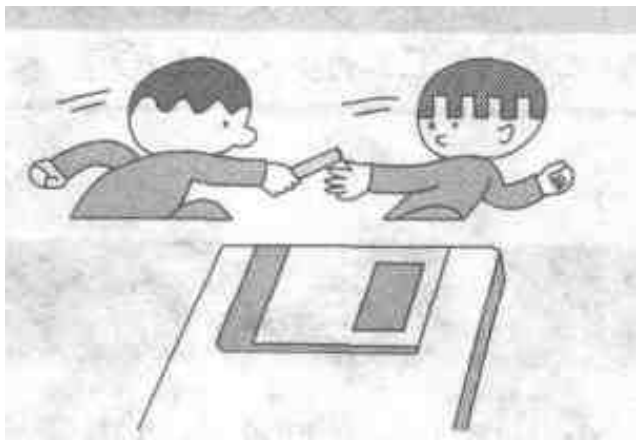
中断的定义为，计算机正在由程序运行的过程中，要通过外部信号来插入另外的运行程序，称为中断。有多个中断时，所用的这一 IC 应当含有优先级别用的中断请求输入端子。

A-D 转换 IC

A-D 转换 IC 是将来自传感器等的模拟信号转换为数字信号的电路所用的 IC，尽管输出的数字信号有并行（并联）与串行（串联）之分，但一般都是用并行（并联）信号。同时，虽然通常用 8 位输出信号作为计算机控制，但为了提高分辨率，也常用更多位数的 A-D。

ADC 0808 等的 ADC08××系列（国际半导体公司），AD7574/7576（模拟器件公司）等，都是各种不同的 8 位 A-D 转换器用集成电路。

练习题



1. RS 232C 是什么标准？
2. GP-IB 是什么标准？
3. 8251 是什么功能的 IC？
4. 8259 是什么功能的 IC？

解答

1. 是串行数据传输用的通信标准。
2. 是并行数据传输用的检测仪器通信标准。与 IEEE-488 标准总线标准相同。
3. 是将 8 位并行数据转换为 8 位串行数据的数据传输通信用的 IC。
4. 是中断控制用的 IC，有 8 只中断输入管脚，根据中断优先级别，向计算机提出中断请求。

6 信号与程序

计算机内外的工作信号是5V、0V的电信号。把5V与0V改为1与0，以（101011100110）形式的数字信号来编制程序，这就称为机器语言。

实际上，不是用机器语言来编程序，而是用与人的语言表示相近的汇编语言或C语言来编程序的。



1 二进制数→十进制数→十六进制数

由于用1与0两个数值表示大的数据时，位数就很多，所以要把二进制数转换为十进制数或十六进制数来使用。

二进制数

3次位	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
2次位	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1次位	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0次位	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
十进制数	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
十六进制数	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

二进制数

只用1与0两个数字记数，全部用完时就向上进位的方法，称为二进制记数法，二进制数的1位数称为1位(比特)。

十六进制数

用0～9与A～F 16种数字，16种全部用完时向上进位的方法称为十六进

制，所表示的数称为十六进制数。

二进制-十进制的转换

人们要用 1 与 0 来编制程序是无法想像的，因此将与十进制相当的二进制数依次用十进制数表示，于是以下的法则成立：

某位是 1 的二进制数，其对应的十进制数是 2 的位号次幂(权)，有多位二进制数为 1 时，对应的十进制数是二进制数为 1 的各个位的权之和。比如：

第 0 位　幂次为 0　$2^0=1$

第 1 位　幂次为 1　$2^1=2$

第 3 位　幂次为 3　$2^3=8$

第 7 位　幂次为 7　$2^7=128$

第 15 位　幂次为 15　$2^{15}=32768$

第 31 位　幂次为 31　$2^{31}=2147483648$

当有多位二进制数为 1 时：

对于 4 位的场合

$(1010)_2=2^3+2^1=10$

$(1111)_2=2^3+2^2+2^1+2^0=8+4+2+1=15$

对于 8 位的场合(参见图 1)

$(10001011)_2=2^7+2^3+2^1+2^0=128+8+2+1=139$

$(11111111)_2=2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0$

$=128+64+32+16+8+4+2+1=255$

对于 16 位的场合

$(1010110001110011)_2=2^{15}+2^{13}+2^{11}+2^{10}+2^6+2^5+2^4+2+1$

$=32768+8192+2048+1024+64+32+16+2+1$

$=44147$

这样，十六位二进制数可用 5 位十进制数来表示，但是当位数很多时，转换计算量也非常之大，所以用下述的十六进制数转换方法。

二进制-十六进制数转换

4 位二进制数可得到 16 种 1 与 0 的组合，因此就相当于 0～9 的数字值与从 A 到 F 的字母，所以可将每 4 位二进制数用以上 16 种数字中相应的记号之一转换为十六进制数。

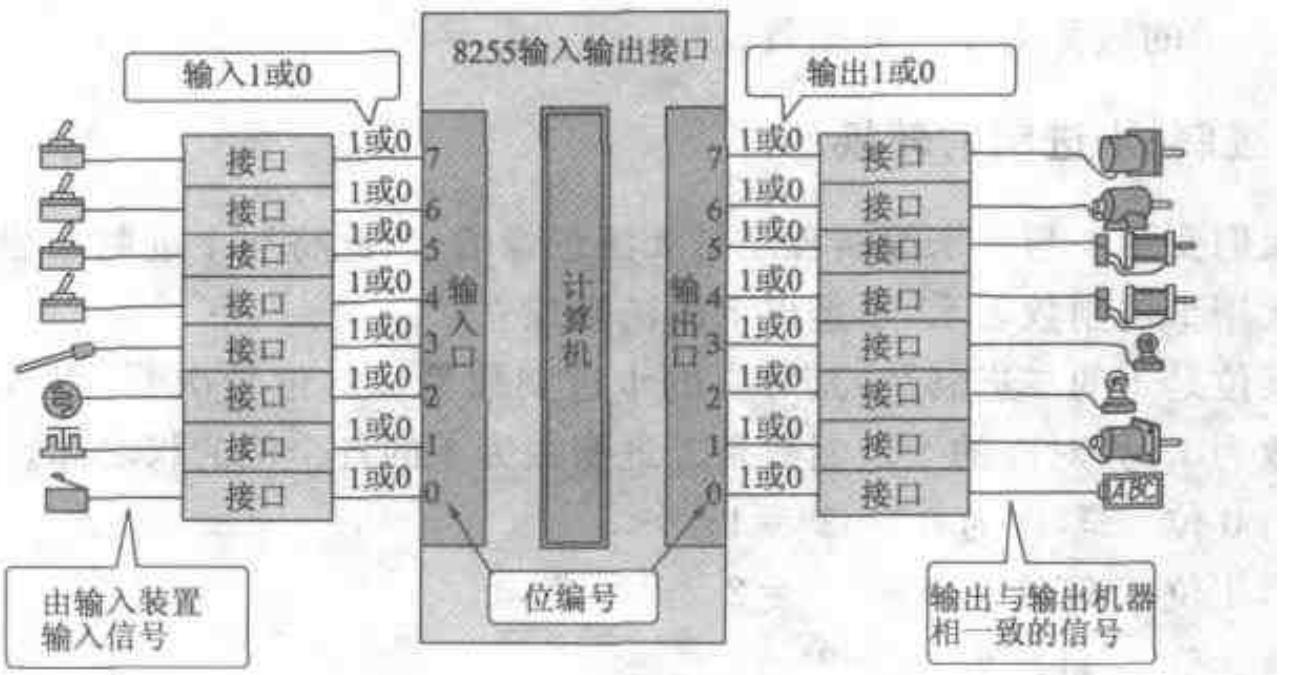


图1　接口与信号

下面给出一些把二进制数转换为十六进制数的例子：

- $(1010)_2=8+2=10=(A)_{16}$
- $(10001011)_2=(1000)_2(1011)_2=(8)(11)=(8B)_{16}$
- $(1010110001110011)_2=(AC73)_{16}$

这样二进制数再增加的话，转换为十六进制数也很简便了。

若知道从第0位到第3位的二进数均为1，其相应权值为1、2、4、8的话，则4位二进制数也可以通过心算算出十六进制数。

计算机控制的场合，与外部连接的各根数据总线都相互独立，可用1(5V)或0(0V)来发出命令或作出判断。在程序命令中，一般用8位或16位的1与0来表示指令，所以用十六进制数处理比较方便。

练习题

1. 用1与0按规则表示的数叫什么数？
2. 为什么要把二进制数转换为十六进制数？
3. 某个位为1时，各自的位值是如何计算的？
4. 把(10100101)，表示为十六进制数是多少？

解答

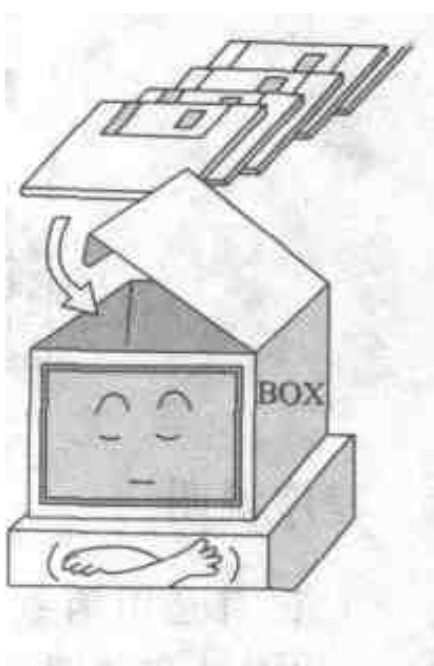
1. 二进制数。
2. 用二进制表示的话，位数太多。
3. 2的位号次幂。
4. $(A5)_{16}$。

7 用于控制的主要程序语言

可以说，没有软件的计算机，只不过是个无用的空机箱，由此可见软件（程序）是多么重要。

程序的本来面目是机器语言，但用机器语言来编制程序是不合适的。要把计算机的各种功能编制为易于理解的语言或函数所组成的程序系统有很多，只要能使计算机工作所用的语言，就称为程序语言。

程序系统是将程序语言编制的程序翻译成机器语言，并变换为电信号存入存储器中。



控制处理用语言主要有汇编语言、C 语言与 BASIC 语言等，这里简单介绍有关控制用的 C 语言。

1 简单的程序例子

尽管这里省略了 C 语言的基本介绍，但对 8255 输入输出接口的各端口，用快速 C 语言编写的输入输出简单程序作出了说明，并给出用 BASIC 语言编写的具有同样功能的程序。

控制动作

当口 A 的 1 号位和 6 号位的管脚输入“1”信号时，从口 B 的 0 号位和 7 号位输出信号“1”。在其他输入信号的情况下，所有位号的输出均为“0”。

输入数据：$(01000010)_2=(42)_{16}$

输出数据：$(10000001)_2=(81)_{16}$

输出数据：$(00000000)_2$ — $(0)_{16}$

C 语言程序

```
1  main ()
2  {
3          unsigned int id;
4          outp(0xd6,0x90);
5          for(; ;)
6          {
7              id=inp(0xd0);
8              if(id= =0x42)
9              {
10                 outp(0xd2,0x81);}
11             }
12             else
13             {
14                 outp(0xd2,0x0);
15             }
16         }
17 }
```

程序说明

C 语言是用函数来组成程序的。并且 C 语言不带 BASIC 程序所用的行号，但此处为了说明方便，还是附加了行号。

1. 函数名 main()是主函数，内容为 2 行到 7 行。

3. 变量 id 为无(+ −)符号的 16 位整型变量，输入输出均不要带(+ −)符号，使用共计 16 位的数据。

用符号";"来结束命令。

4. outp()是输出命令函数。

CW 寄存器(地址$(D6)_{16}$)中，把 CW 数据$(90)_{16}$作为输出命令，地址或数据用十进制数、十六进制数或变量均可，用十六进制数时，要在前面加 0x。

5. 无条件跳转(循环控制)语句。

无条件地在第 6 行与第 16 行之间进行无限次反复执行(实际上是需要有跳出循环圈的程序的)。

7. inp()是输入命令函数。

在变量 id 中代入 I1 A(地址$(D0)_{16}$)的输入数据。

8. 条件分支控制语句。

当变量 id 的值等于$(12)_{16}$时，执行第 9 行与第 11 行之间的命令，其他情况下均执行 else 后面的第 12 行与第 14 行之间的命令。

10. 14 是输出命令函数，输出 I1 B(地址$(D2)_{16}$)上的各个数据(参见图 1)。

BASIC 语言的例子

与 C 语言同样功能的 BASIC 语言程序如下。

```
10  OUT  &HD6, &H90
20  I=INP (&HD 0)
30  IF  I=&H42  THEN 40     ELSE70
40  OUT  &HD2,&H81
50  GOTO  20
60  OUT  &HD2,&H 0
70  GOTO  20
80  END
```

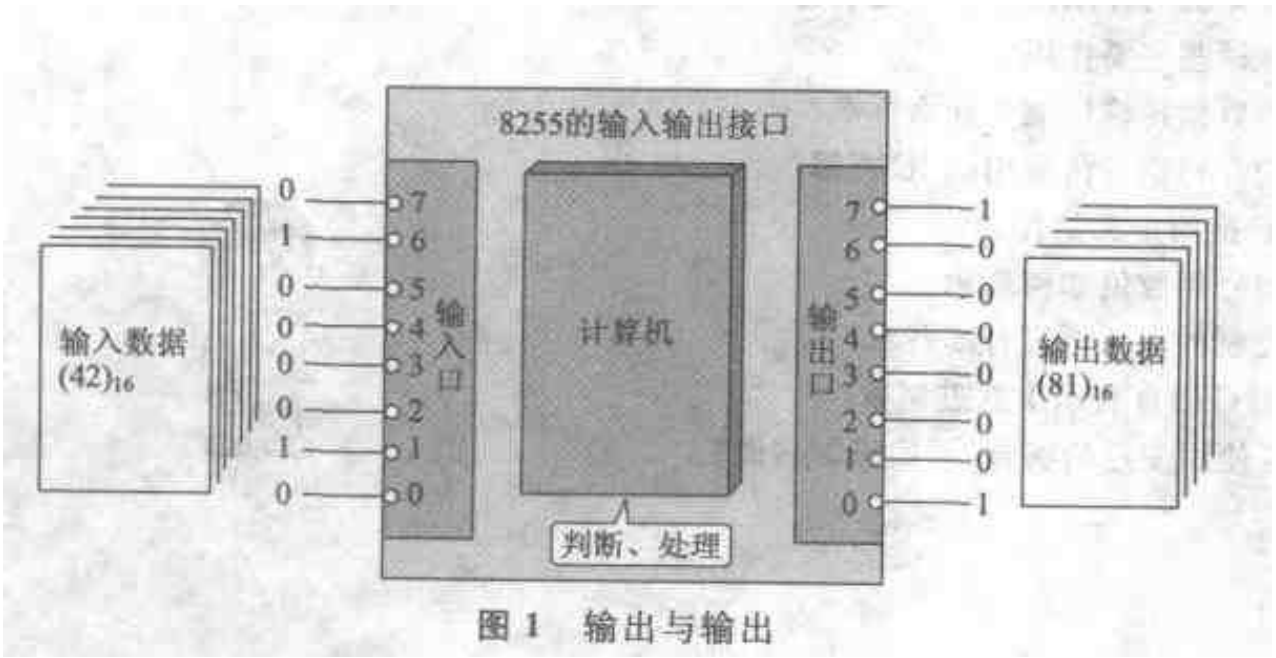


图 1 输出与输出

练习题



1. C 语言如何标记十六进制数的数值？
2. 什么是把数据往指定输出口输出的函数？
3. 什么是把数据往指定输入口输入的函数？
4. for(;;)是什么函数？

解答

1. 前面带 0x。
2. outp()函数。
3. inp()函数，把输入数据代入变量中。
4. 无继续条件时的 for 语句，在{}之间执行无限次循环。

Q 挑战选择题

① 计算机是由什么构成的?
② 计算机的心脏常称 CPU,由什么组成?
③ 计算机内外信号的传输线是什么名称?
④ 计算机的控制信号是什么信号?
⑤ 个人计算机的各种总线用什么方法与外部连接?
⑥ PC 9801 扩展槽的数据总线是多少位的?
⑦ PC 9801 扩展槽用 5V 时每个槽是用多少安[A]的电流?
⑧ 用计算机处理数据信号的电流大约是多大?
⑨ 计算机的处理时间是以什么为基准的?
⑩ 调整计算机与外围设备信号的装置叫什么名称?
⑪ 接口有哪些主要作用?
⑫ 通用的数据传输标准是什么标准?
⑬ 常用的并行信号传输用的 IC 有哪些?
⑭ A/D 转换的定义是什么?
⑮ 数字信号用数值如何表示?
⑯ 用于控制的主要语言有哪些?
⑰ 如何用 C 语言表示十六进制数?
⑱ C 语言使用变量的场合,开始该怎么做?

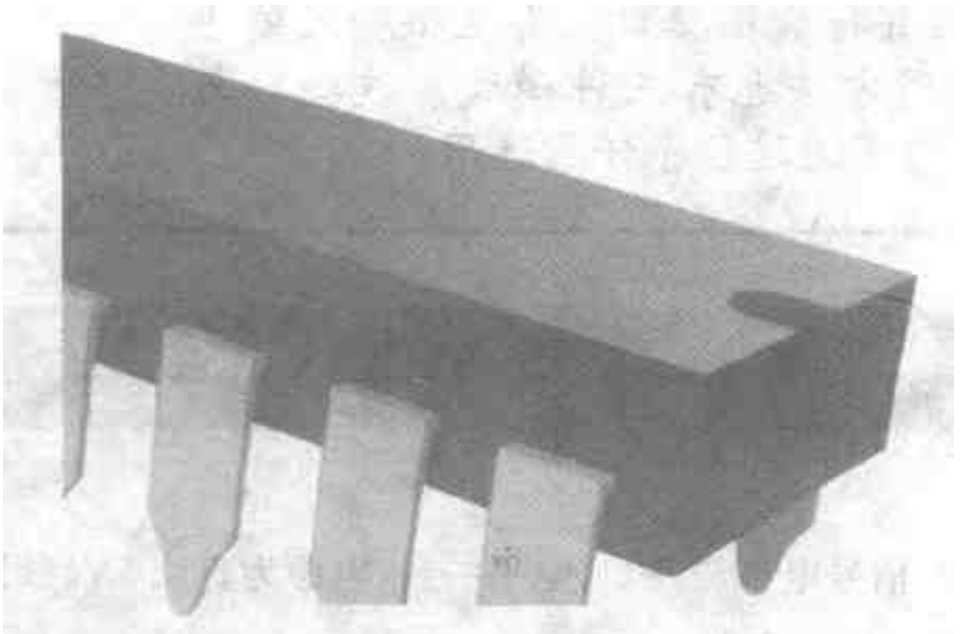


A 答　案

①命令控制单元、算术逻辑运算单元、存储装置、输入输出装置等等。　②由称为处理器或处理单元、命令控制单元与算术逻辑运算单元构成。　③总线,有数据总线、地址总线、控制总线。　④是电压为 5V 与 0V 的数字信号。　⑤从扩展的总线连接引出。　⑥是 16 位,32 位个人计算机,也用低端 16 位。　⑦是 0.8A,只用于接口电路等的电源。　⑧与 TTL 的电平一致,是从数毫安到数十微安的微小电流。　⑨是计算机用的时钟周期。　⑩称为接口。　⑪电流放大或交直流变换等电气条件的调整。　⑫是 RS-232C 标准、GP-IB 标准等。　⑬是 8255。　⑭是将模拟信号转换为数字信号用的。　⑮用电气信号 5V 表示"1",0V 表示"0",再用十进制数或十六进制数表示。　⑯汇编语言、C 语言、BASIC 语言等。　⑰数值的前面加 0x。　⑱要先进行类型声明。

第6章

输入输出装置与接口



制作接口的好方法

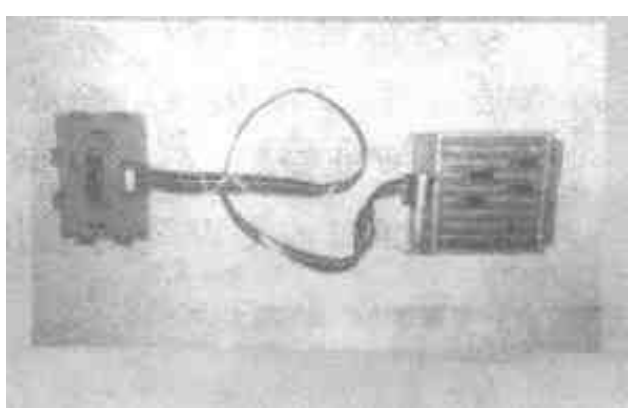
要启动电动机等的场合，操作者用手指操作开关，启动电动机。机械是否正常运行着，要靠耳闻目睹来判断机械的状态。

那么，计算机是怎样使接点动作、启动电动机，使灯亮起来，还要检测周围的状态呢?

本章将介绍关于计算机控制基础的有关知识，即把开关或传感器的信号输入计算机，为了操作机械所需的信号的输出等方法，以及计算机的基本输入输出设备与有关接口等知识。

我们已经学习过，计算机的各种总线不仅可以直接连接到输入输出装置上，还可以通过接口来连接。这里介绍一下通过8255连接的输入输出接口，各种输入输出装置专用接口的连接方法。

8255是最常用的输入输出接口，它有三个输入输出口，对应状况的使用条件可通过程序来变更。为了理解这种接口，让我们从制作接口板开始，一直到利用程序进行最终检验，学习全部过程。同时学习一些与接口有关的计算机基本知识，如用开关或传感器，把信号输入到计算机，把信号输出到电动机等执行装置，或控制指示灯所用的接口上，制作用个人计算机处理的简单的输入输出接口等。



1 8255输入输出接口管脚与使用模式

8255是用8086系列微处理器设计的最常用的8位并行数据接口，有三个输入输出口。其使用方法也有三种模式，这些口与使用模式均可通过程序任意选用。

I　8255 IC 的管脚

8255 共有 40 只管脚，信号电平与 TTL 电平一致，电源为直流 5V(参见图 1)。

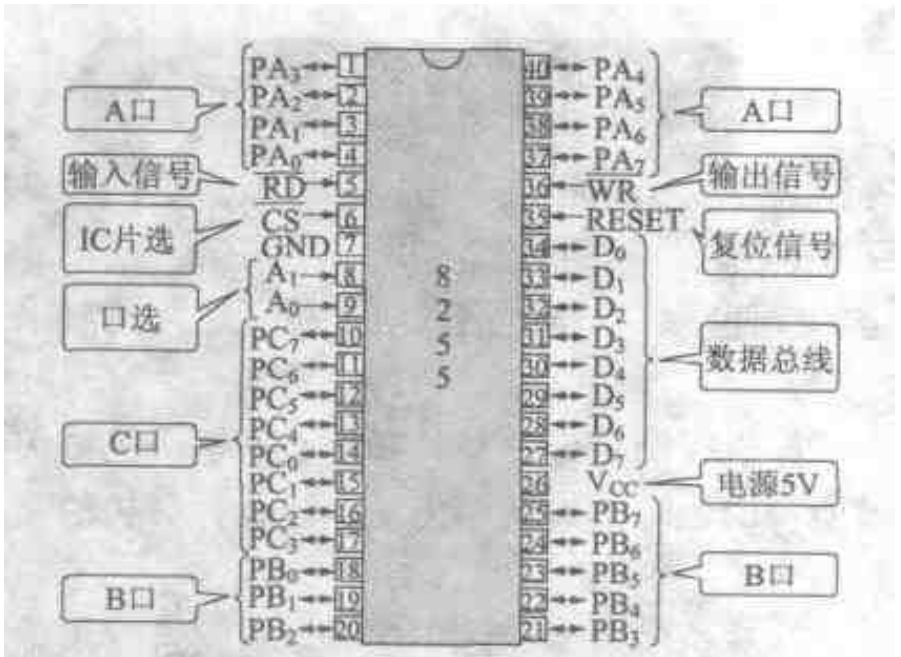


图 1　8255 的管脚

① 数据总线管脚($D_0 \sim D_7$ 8 只)

与计算机端的 8 位数据总线连接，是进行数据传输的 8 只管脚。

② 片选管脚(CS)

是选择 8255 集成电路芯片的信号输入管脚，以“0”信号驱动。

③口地址管脚(A_0，A_1 两只)

是选择口或 CW 寄存器用的信号输入管脚。

④ **读入数据信号管脚**(RD)

一发出输入命令,此脚上就输入"0"信号,就将指定口的输入数据通过数据总线传输到计算机中。

⑤ **写数据信号的管脚**(WR)

一发出输出命令,此脚上就输入"0"信号,就将来自计算机的输出数据输出到指定的口上。

⑥ **复位管脚**(RESET)

若此脚输入"1"信号,则 IC 片内的寄存器就清零。

①～⑥管脚均与计算机连接。

⑦ **电源管脚**(V_{CC}、地)

IC 的驱动电源,有电源正极管脚(V_{CC})与接地管脚(GND)。V_{CC}管脚为直流 5V。电源最大使用电流为 120mA。与电源连接时,必须注意电压是否稳定与干扰情况。同时,为了使信号电压的电平一致,计算机与外围设备的接地线必须连接,但采用光电隔离型的场合,不必连接。

⑧ **输入输出口管脚**

数据输入输出用的管脚是 A 口(PA_0～PA_7)、B 口(PB_0～PB_7)、C 口(PC_0～PC_7)各 8 只,共 24 只。C 口有时也作为控制信号管脚用。

Ⅱ　口的使用方式

口有模式 0、模式 1 与模式 2 三种使用方式,通过设定 CW(控制字)来选择口的使用方式,并把它输出到 8255IC 的 CW 寄存器中。

① **模式 0**

这是基本的使用方式,将 A 口、B 口、C 口设定为输入口或输出口。不能把一个口既作为输入口又作为输出口用。C 口的高四位与低四位则可分别使用。

可将输出数据保持(锁定)在口寄存器中,直到有下一个数据到来为止。输入数据则不予保持。

② **模式 1**

这是单向联络模式,在数据的传送过程中,联络上控制信号,就可取得数据输入输出的延时(参见图 2)。

将 A 口与 B 口设置为输入或输出口,用 C 口的 PC_0～PC_2 三位作为 B 口的控制信号,PC_3～PC_7 作为 A 口的控制信号(参见图 2 和图 3)。

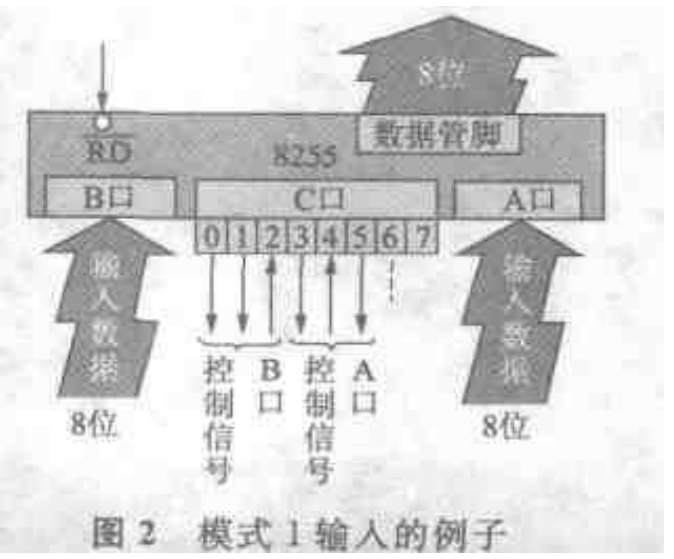


图 2　模式 1 输入的例子

② 模式 2

这是双向联络方式，将 A 口作为输入输出口，将 C 口的高端开始的 5 位作为控制信号使用(参见图 4)。

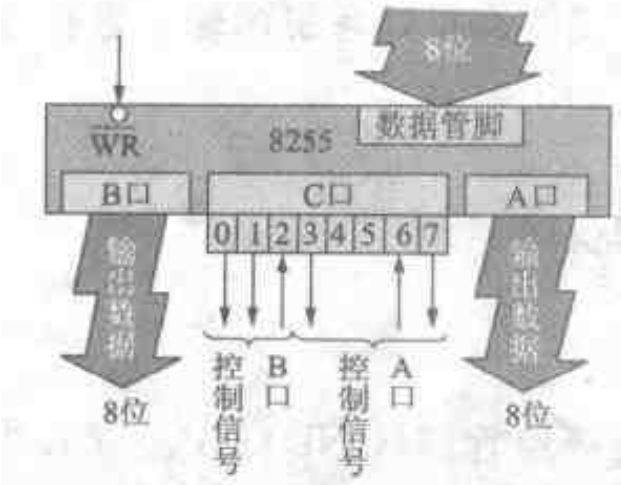


图 3　模式 1 输出的例子

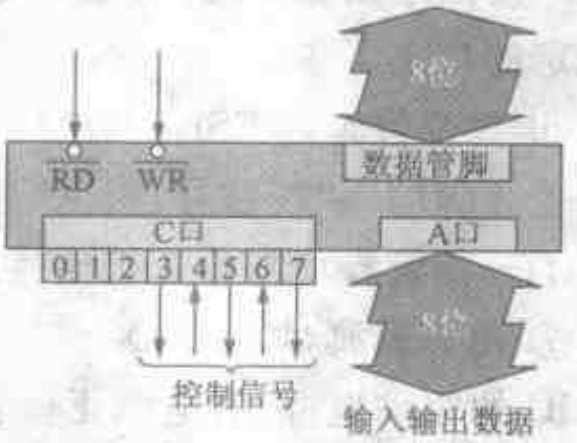


图 4　模式 2 的输入输出例子

练习题

1. 8255IC 的电源是几伏？
2. 与计算机端连接的数据线有多少根？
3. 模式 0 中，将口的数据输出的场合，在下一数据输出之前，口的管脚状态是什么样的？
4. 模式 1 的联络控制信号是从哪个口输出的？

解答

1. 是直流 5V，最大电源电流标准是 120mA 左右。
2. 因要交换并行 8 位数据，所以是 8 根总线。
3. 输出口中一直保存数据，到下一个数据输出之前，该数据一直处于输出状态。
4. 是 C 口。

2 试制8255输入输出接口板(1)

在个人计算机PC9801扩展槽中，插入8255接口板，并制作与外部机器连接输入输出口用的接口板，两块板之间用28线扁平电缆连接。

Ⅰ 个人计算机与连接条件

① 数据总线采用扩展槽的数据总线管脚的低端 8 位。

② 地址总线采用扩展槽的地址总线管脚的低端 8 位。

因使用低端的 8 位数据、地址成为偶数号单元。用户能开始使用的地址设定如下：

A 口：$(D_0)_{16}=(11010000)_2$　　B 口：$(D_2)_{16}=(11010010)_2$

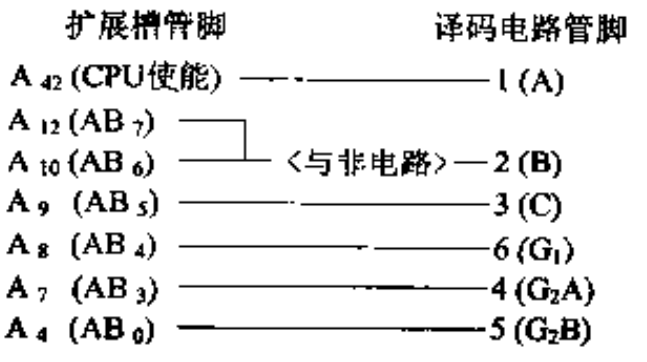
C 口：$(D_4)_{16}=(11010100)_2$　　CW 寄存器$(D_6)_{16}=(11010110)_2$

Ⅱ 8255 接口板

在 PC 9801 扩展槽专用的通用印制电路板中，对 8255 译码电路、与非门电路等 IC 集成电路进行配线。

① 有关地址的连接

在扩展槽地址总线的低端 8 位中，把可变化的 AB_1 与 AB_2 和 8255 端的口地址选择管脚 A_0、A_1 相连接，即可选择指定的口。



将低端其余 6 位地址总线与 CPU 使能(ENB)信号接到所用的译码电路(74 LS 138)及与非电路(74 LS 00)上，其对应的连接关系如下：

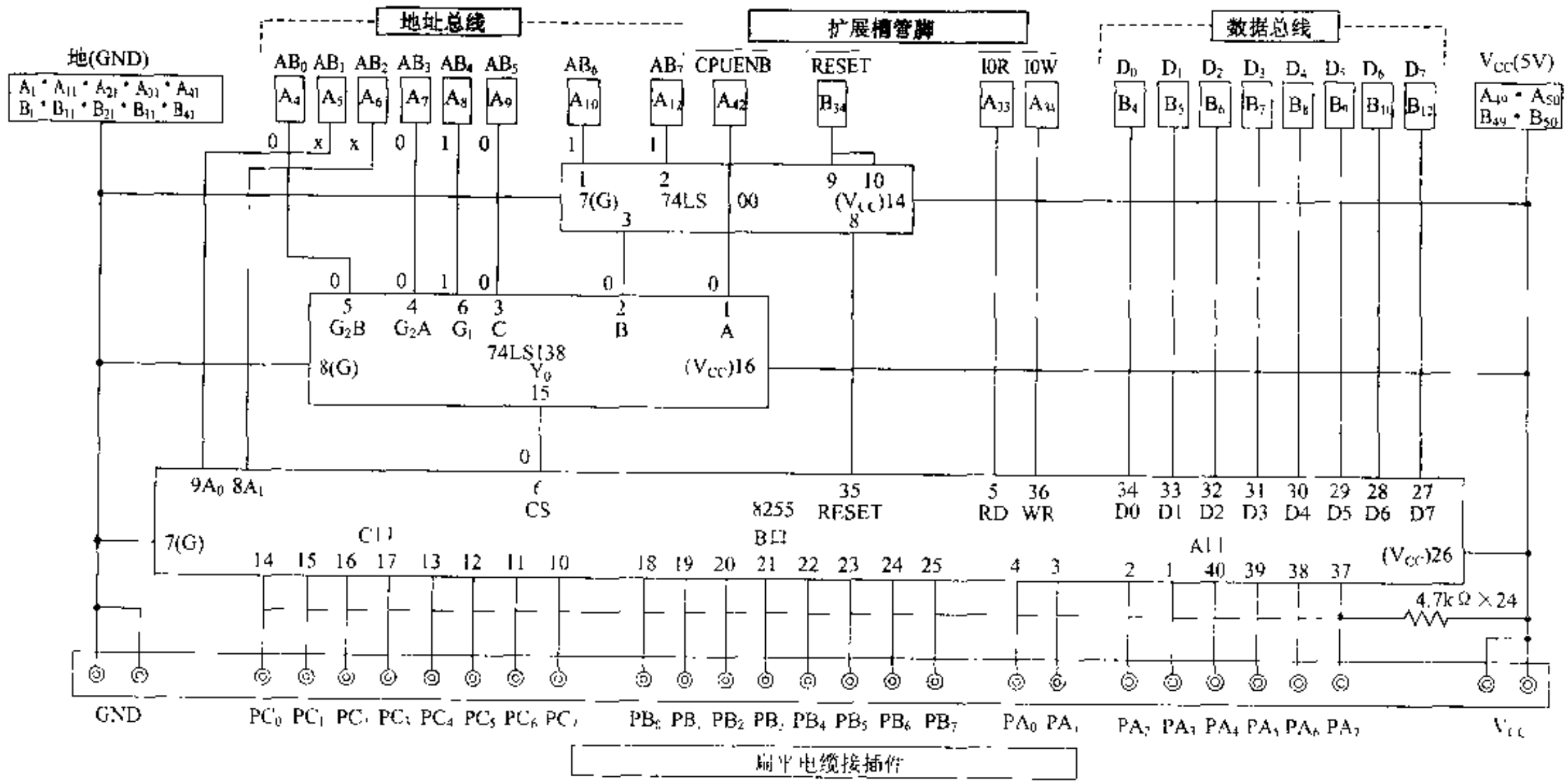


图 1 8255接口板电路

然后，将译码电路的输出管脚 Y_0 连接到 8255 的 CS(片选)管脚(低电平有效)。

当端口地址 $(11010XX0)_2$ 上，CPU 使能信号(ENB)为"0"时，译码电路输出脚 Y_0 的输出信号成为"0"，输入到 CS 端，以驱动 8255 集成电路。CPU 的使能信号，是观察处理单元(处理器)是否占用数据总线的控制信号。

② 数据总线

扩展槽 16 位数据总线的低端 8 位连接到 8255 的 8 位数据总线管脚上。

③ 复位信号的连接

因计算机端的复位信号为低电平有效，而 8255 端的复位信号为高电平有效，因此复位管脚之间的连接要在中间接入反相电路(①中的与非电路)。

④ 写(WR)管脚、读(RD)管脚的连接

将扩展槽的 IOR(读)管脚与 8255 的 RD 管脚连接，IOW(写)管脚与 8255 的 WR 管脚连接。

⑤ 电源的连接

8255IC 的电源使用扩展槽的电源，5V 电源线的 4 只管脚与地线的 10 只管脚均设置在各自的基板上，为防止噪声干扰，在 IC 电源管脚附近与扩展槽的连接电源的端子附近接入电容器。

⑥ 口数据总线(参见图 1)

8255 的各输入输出口管脚，用扁平电缆与外部的输入输出口所用的板连接。

为便于连接放大电路，将 4.7kΩ 的电阻接在电源与各数据总线上(上拉电阻)。

⑦ 扁平电缆接插件管脚

将数据总线与输入输出板连接。

练习题

1. 为什么口地址是偶数号？
2. 为什么要将扩展槽地址总线的 AB_1，AB_2 接到 8255 的 A_0 与 A_1 上？
3. 为什么使用 CPUENB 信号？
4. 为什么要在 IC 的电源管脚等处接入电容器？

解答

1. 是从 32 位或 16 位数据总线的低端 8 位得到的地址。
2. 为了改变用二进制数表示的偶数号口地址 A_1 与 A_2 的值。
3. 为了用来自处理单元的命令确认是否在使用数据总线。
4. 为了吸收噪声。

3 试制8255输入输出接口板(2)

1 输入输出接口板

这是将外围设备连接到带 A 口、B 口与 C 口的输入输出接插件的接口板。输入输出接插件分为高位管脚和低位管脚，每一接插件有口数据管脚 4 只，5V 电源管脚及接地管脚共 6 只。配置情况如图 1 所示。因为板上没有信号放大，

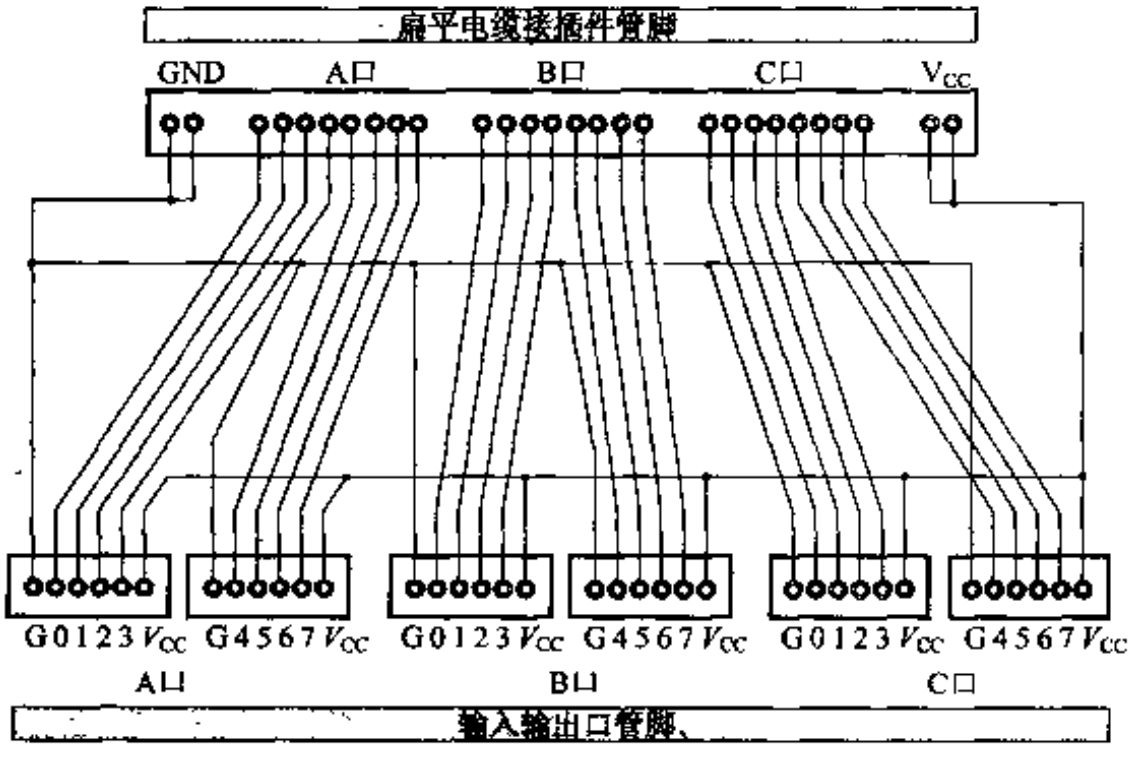


图 1　输入输出用接口板

所以由口接插件到外围设备的连接线应当尽可能短，如果需要连接线长到某一长度的话，则必须带放大电路。

5V 电源的一个扩展槽，最大只能用 0.8A 的电流，这一电流是供给控制电路或信号放大电路用的，外围设备最好使用其他的电源。

Ⅱ　8255 输入输出板的检验

若没有检验 8255 输入输出板的工作是否正常的话，是不能使用的，否则因错误的接线会造成损坏个人计算机的大事故。

焊点的检验

使用放大镜等对焊点是否合适，是否接触到其他管脚等进行检验，焊接不良的焊点应当补焊。

短路的检验

检验电子器件管脚或配线有无短路。短路的原因往往不明显，检查时必须多注意，尤其要彻底检查电源有无短路，电源不仅有 5V 的，还有 12V 的。

配线检验

是否已按配线图配线，要用万用表等进行检验。虽然麻烦，但必须有耐心地做好检验工作。尤其要注意 IC 的管脚有否搞错。

插座管脚面的检验

对插入扩展槽的管脚面上的焊点是否饱满，有无焊油等污染进行检验，并进行清理等，使管脚面整齐、美观。绝缘不良往往是产生噪声的根源，所以必须从里外两面进行检验。

用程序来检验功能

将接口板安装到个人计算机的扩展槽中，用程序进行输入输出检验。

在 A、B、C 各口上，用从$(0)_{16}$到$(FF)_{16}$为止的数据，依次输出，将当时口的输入数据与输出数据相比较，若全部相同的话，则可判断为功能正常。

同时，也可以对连接在各口上的输入输出装置，如发光二极管或开关等进行检验。

检验程序

```
/*8255接口板的硬件检验*/
#include〈stdio.h〉
main (  )
{
    unsigned int a, aid, bid, cid,od;
    printf ("￥x 1b ［2j"］ );
    outp (0xd6, 0x80);
    for (od=0x0;od〈=0xff;od+=0x1)
     {
        outp(0xd0,od);aid=inp(0xd0);
        outp(0xd2,od);bid=inp(0xd2);
        outp(0xd3,od);aid=inp(0xd3);
        printf("输出%X￥n"，od);
        printf("输入%x%x%x￥n"，aid,bid,cid);
        if(aid==od‖bid=od‖cid==od)
        {
        }
        else
        {
            printf("I/O错误￥t");
            goto brk;
        }
    }
    printf("￥nOK￥n");
brk:;
}
```

练习题

1. 用输入输出接口板进行信号放大的场合，使用什么电路？
2. 吸收噪声用的电容器接在什么地方？
3. 制作好的电路板，必须进行哪些检验？
4. 用程序检验能检验什么功能？

解答

1. 可用通过接到电源的电阻与数据总线连接的上拉电路。
2. 接在扁平电缆接插件的电源附近。
3. 应对焊点是否良好、短路(特别是与电源)、接线错误、插座的管脚面等进行检验。
4. 对各口上的输入输出进行检验。

4 8位发光二极管的点灯电路

发光二极管的点灯电路用于确认8位口的各位所输出的数据。

先考察一下用来自8255信号的发光二极管点灯电路。

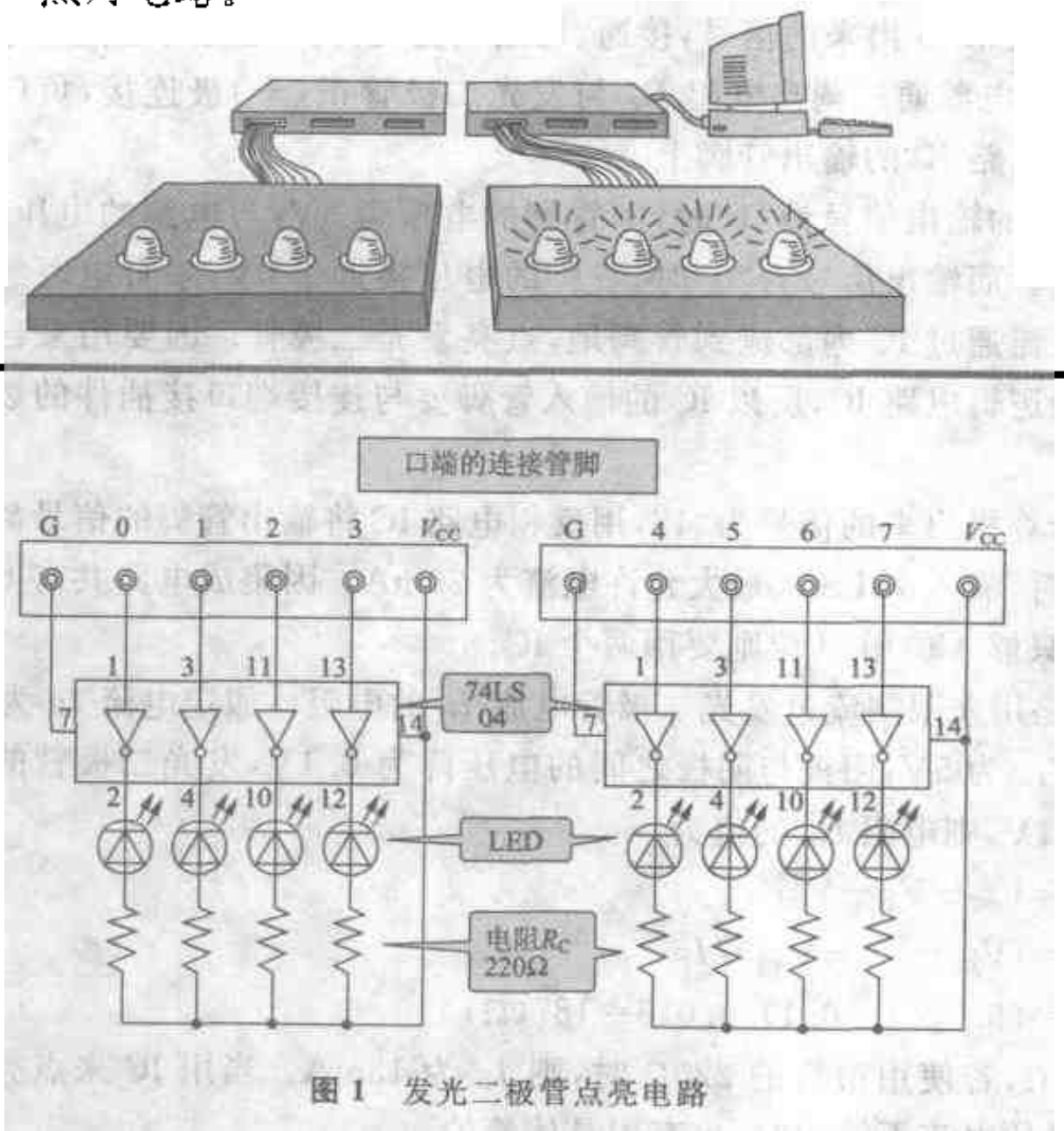


图1　发光二极管点亮电路

所用发光光二极管的主要规格

型号：TLR113A(东芝)GdP(磷化镉)

最大电流：I_F=20mA。

标准正向电压：V_F =2.1V。

推荐工作电流：I_F =10～15mA。

发光强度(轴向)=3.5mcd(I_F=15mA)。

信号"1"时的输出口的电流

由于上拉的电流约为 1mA，来自 8255IC 的电流最大为 4mA 的话，则 8255 接口板上的信号为"1"时的输出电流为 5mA。但要让发光二极管点亮的电流需要 10～15mA 左右，所以不能将信号电流直接用于亮灯。

使用逻辑电路 IC 的发光二极管电路(参见图 1)

利用从 8255 出来的信号，接通、切断电流的开关上使用逻辑电路 IC。如图 1 所示，5V 电源通过调整电阻 R_c 与发光二极管正(+)极连接，负(-)极则连接到逻辑电路 IC 的输出管脚上。

当 IC 的输出信号为"1"时，则管脚的电压为 5V，与电源的电压相等，没有电流流过。而输出信号为"0"时，管脚的电压接近于 0V，来自电源经过发光二极管的电流通过 IC 内部流到管脚地，点亮发光二极管。因要用来自计算机的信号驱动逻辑电路 IC，所以 IC 的输入管脚要与连接端口接插件的数据管脚相连接。

从计算机出来的信号为"1"，用反相电路 IC 将输出管脚的信号翻转为"0"。"0"信号时，流入 74LS04 最大允许电流为 20mA。因集成电路共有 6 只输入管脚，所以只够 4 位用，8 位则要用两个 IC。

R_c 是用来限制流过发光二极管电流 I_F 的电阻。假定电流 I_F 为 15mA，电源电压 V_{CC} 为 5V，阳极与阴极之间的电压降为 0.1V，发光二极管的发光电压 V_F 为 2.1V，则电阻 R_c 的值为

$$V_{cc}=V_{cc}-V_F-V_{ad}$$

$$R_c=(V_{cc}-V_F-V_{ad})/I_F$$

$$=(5-2.1-0.1)/0.015=187(\Omega)$$

需要 187Ω，若使用市售的 220Ω 时，则 I_F 为 13mA。当用 IC 来点亮发光二极管所流过的电流不够用时，也有用晶体管的。

简单的亮灯程序

C 语言是不带行号的，下面给出的程序中加上行号是为了便于说明。

```
1  main ()
2  {
3        unsigned int n;
4        outp(0xd6,0x90);
5        for(; ;)
6        {
7           printf("亮灯数据n=");
8           scanf("%x",&n);
9           outp(0xd2,n)
10       }
11 }
```

程序说明

这是由键盘输入的十六进制数来使发光二极管点亮的程序。程序中各行的意义如下：

3. 变量 n 是无符号整型变量。

4. 将 CW(控制字)数据$(90)_{16}$输出到 CW 寄存器(地址$(D6)_{16}$)中的命令。设置 A 口为输入口，B 口与 C 口为输出口。

5. 执行第 6 行到第 10 行间的无限循环。

7. 显示屏上的“亮灯数据 n=”。

8. 通过键盘，将十六进制的亮灯数据输入到变量 n 中。

9. 将键入的变量 n 之值，输出到 B 口(地址$(D2)_{16}$)。这是用于发光二极管亮灯电路输出练习用的很方便的电路。

练习题

1. 使发光二极管亮灯要多大的电流？信号电流可以亮灯吗？
2. 使发光二极管发光的电压是多大？
3. 发光二极管电路的电阻是干什么用的？
4. 发光二极管亮灯电路的反相电路有何作用？

解答

1. 根据种类不同而不同，约 10～20mA 左右。
2. 根据种类不同而不同，约 2V 左右。
3. 限制流经发光二极管的电流。
4. 接通、切断流经发光二极管的电流。

5 开关用接口电路

开关用接口电路的定义是，将开关的输入信号变换为计算机能处理的信号的输入电路。

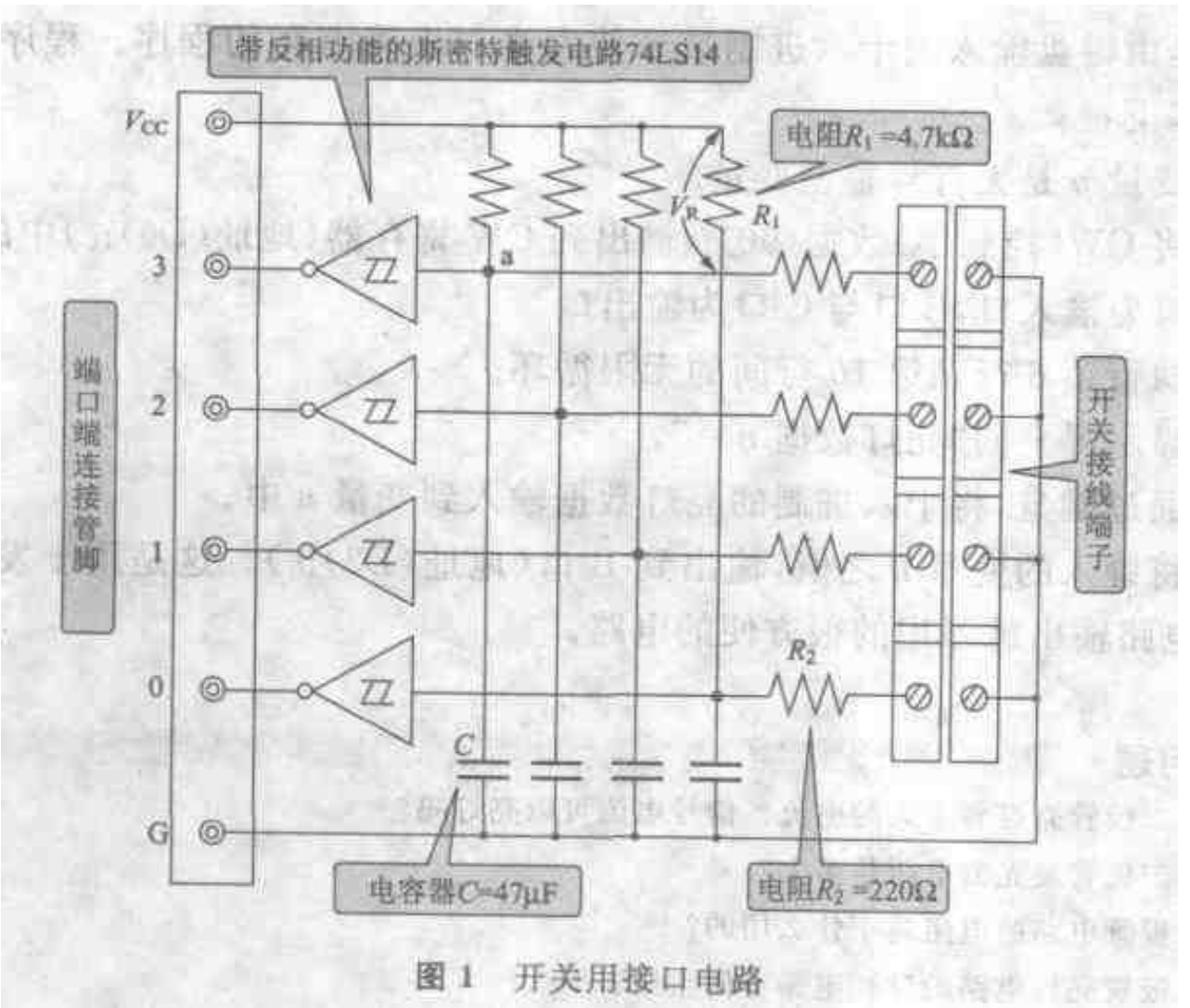


图1　开关用接口电路

Ⅰ 振　动

按钮开关或限制开关等，由于弹簧的复位功能而在开始闭合或断开时，会产生约10ms的反复通断的振动。

要消除振动的影响，要用电路将波形的振动部分进行整形，或考虑用程序来去除振动。例如开关用接口电路，是用波形整形电路来消除振动，整形为美

观的波形。

尽管有各种波形整形电路，但这里只学习一种由积分电路和斯密特电路所组成的波形整形电路，前者使振动的波形变得平滑，而后者则将这种平滑的波形变成方波。

Ⅱ 积分电路

电阻 R_1 与 R_2、电容 C 所构成的电路，通过接点使振动的波形变得平滑。

打开接点

电流从电源通过电阻 R_1 流入电容 C，使 a 点电压逐渐升高，此时振动被消除，同时使波形平滑。当电压上升到饱和电压的 63%时，所需的时间称为时间常数，它由电阻 R_1 与电容器的电容 C 的乘积决定。一般时间常数规定为比振动时间长。

电容器充电饱和后，电流就流入 IC，该电流非常小，$I_{IH}=0.02\text{mA}$，电阻上的压降 V_R 也非常小，信号电压为“1”。

闭合接点

电路中采取接地措施，电流从 5V 电源流经电阻 R_1 与电阻 R_2 入地，电容器放电的电流加到从电源流出的电流，而使振动波形变为平滑的波形。放电结束时，应当使 IC 输入管脚的电压成为“0”信号的许用电压以下，以此来确定电阻 R_1 与 R_2 的值，电阻 R_1 与 R_2 之比就是电压之比。

时间常数

用电阻 R_1 来限制流经电容 C 的电流，通过电容与电阻来调整电容器的饱和时间。已经学过的使电压上升到 63%饱和电压的时间 τ 称为时间常数，而在放电场合，则使电压降到 37%饱和电压的时间作为时间常数。

按以上条件规定时间常数 τ，应当大于振动的时间，以此条件来计算电阻值和电容值。

假定时间常数 τ 为 10ms(0.01s)。

$$\tau=C\times R=0.01$$

$$C=\tau\div R=0.01\div R$$

充电时取 R 为 4.7kΩ，则

$$C=0.01\div 4700=2.1(\mu\text{F})$$

放电时，取 $R_2=220\Omega$，则

$$C=0.01\div220=45.5(\mu F)$$

应取大一些的值，用 $47\mu F$ 的市售电容器。

Ⅲ　施密特触发电路

它将平滑变化的输入电压信号整形为电压高低分明的方波信号：当电压上升到 1.6V 以上时，作为“1”信号输出；当电压下降到 0.6V 以下时，作为“0”信号输出。同时使用当接点闭合时输出“1”，接点断开时输出“0”，具有反相功能的 74LS14 集成电路。

Ⅳ　接线端子

接线端子处设置为 4 位的开关接口。就计算机端来讲，用与其连接用的输入口的 4 位信号端子、连接 5V 电源接地用端子、用的端子共 6 只接插件管脚。5V 电源是开关与集成电路用的电源。

开关端的连接是用螺钉固定式的接插件，电阻 R_2 端的接线端子与接地端的接线端子各一对，共 4 处。

练习题

1. 什么是振动？
2. 修正振动波形的电路有哪些？
3. 什么是电容的时间常数？
4. 施密特触发电路是干什么用的？



解答

1. 振动就是短时间内，信号电压忽高忽低反复振动。
2. 修正振动波形的主要方法，是用积分电路和施密特触发电路。
3. 到达饱和电压的 63%所需的时间。
4. 把平滑的波形变为高、低分明的波形。

6 电磁继电器与接口

此接口使用计算机输出信号使电磁继电器动作，以继电器接点来接通、切断别的电源。是用于对控制对象进行控制的接口。此时的信号占用4位(bit)信号位。

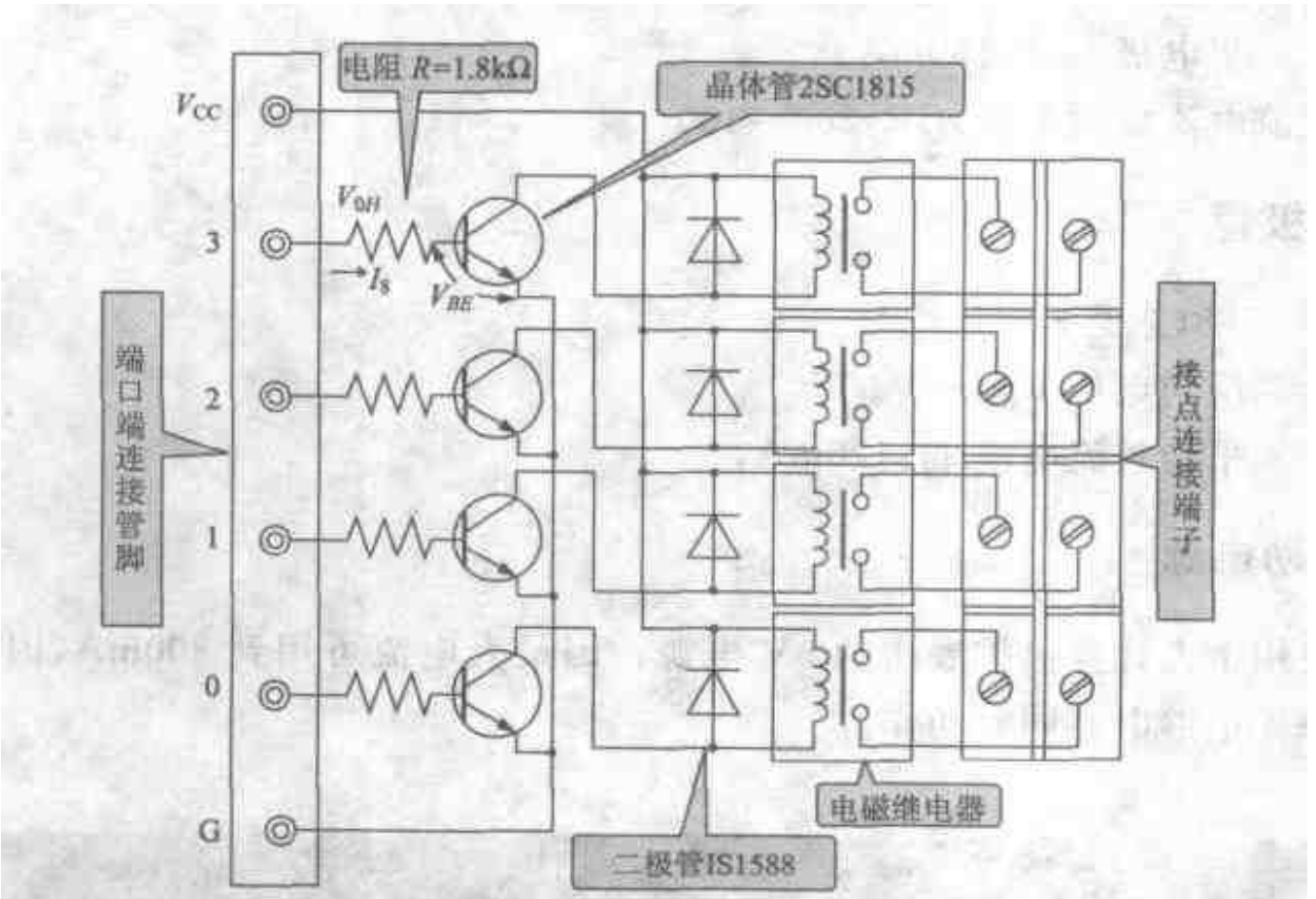


图1 电磁继电器接口

I 使用器件的主要规格

电磁继电器规格

型号：HB2-DC5(松下)。

线圈额定电压：直流 5V。

励磁电流：72mA。

接点规格：转换接点 2 个。

接点容量：2A-125V　AC。

　　2A-30V　DC。

动作/复位时间：约 5ms。

最大操作频率：20 次/s。

晶体管规格

2SC1815(NPN 型)GR 型

最大定额：

集电极-基极之间电压：60V。

集电极-发射极之间电压：50V。

集电极电流：150mA。

直流电流放大系数：GR：200～400。

二极管

IS 1588(东芝)

反向电压 V_R：30V。

平均整流电流 I_0：120mA。

驱动电源

使用个人计算机扩展槽的 5V 电源。因最大电流可用到 800mA，因此，规定允许 4 个继电器同时动作。

Ⅱ　电　路

电磁继电器电路

这是一种通过晶体管的电流来驱动继电器的电磁线圈的电路。

5V 电源→继电器线圈→晶体管集电极→发射极→地，在线圈两端反向并联吸收反电势(浪涌电压)用的二极管。

信号电路

通过插接件的数据管脚引出的保护电阻 R 与晶体管的基极连接。

因集电极电流 I_C 最大为 150mA，所以可根据直流电流放大系数 $h_{fe}=200$

计算出最大的基极电流为 0.75mA。

这一基极电流 I_B，经电阻 R 降压后，基极电压应大于导通电压(0.7V)，据此可求得电阻值 R。

$I_B = I_C / h_{FE} = 150\text{mA} \div 200 = 0.75\text{mA}$

$R = (V_{OH} - V_{BE}) / I_B = (2.4\text{V} - 0.7\text{V}) \div 0.75\text{mA} = 2\ 267\Omega$

电阻应当用小一些的 1.8kΩ 市售电阻。V_{OH} 以"1"输出的最小电压值来计算。

端　子

继电器接点的 a 接点端子连接在螺钉固定的接插件上，作为与外部连接的管脚。

11　简单的动作程序

```
1  main( )
2  {
3        unsigned int i;
4        outp(0xd6,0x90);
5        for(;;)
6        {
7           i=inp(0xd0);
8           if(i==128)
9           {
10                break;
11          }
12          else if(j<=128)
13          {
14                outp(0xd2,i)
15          }
16       }
17 }
```

在 A 口上，连接 8 个开关输入电路，B 口的低 4 位连接电磁继电器的接口。

若按下低 4 位的开关，则所编的程序就使同一位号的继电器动作，若按下第 7 位的开关，就停止执行程序。

练习题

1. 与电磁继电器线圈并联的二极管起什么作用？
2. 为使晶体管导通，需多大的基极电压？
3. 在晶体基极的前面接入的电阻起什么作用？



解答

1. 为了吸收反电势(浪涌电压)。
2. 0.7V 以上。
3. 为了保护晶体管。

7 小型直流电动机的接口



制作能控制根据计算机发出的转向与启动、停止控制两个信号，使小型直流电动机正反转或停止的接口。

所接小型直流电动机的额定电压为直流12V，额定电流为500mA。

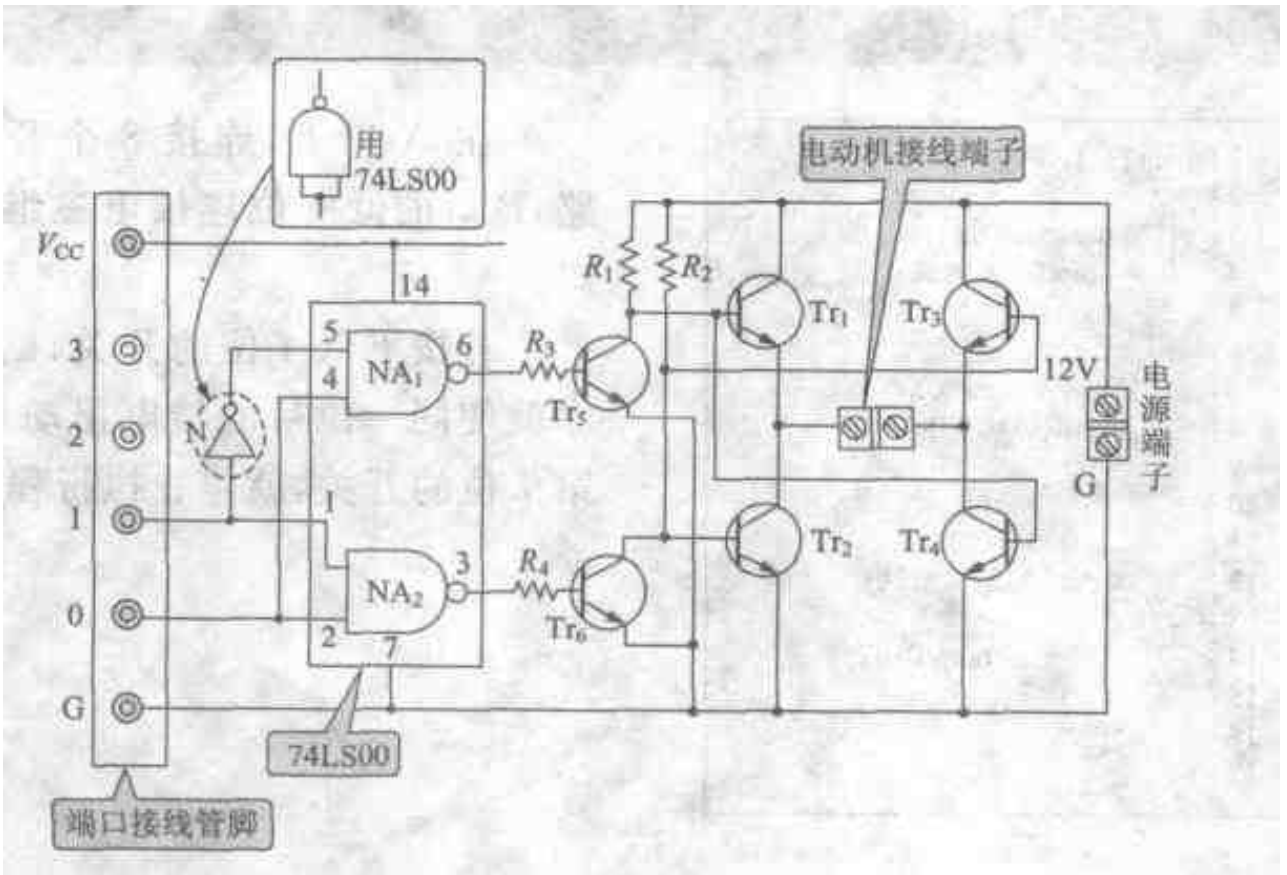


图 1　直流电动机的接口

I 电　路

与计算机的连接

与计算机端用 6 脚的接插件连接，转向与启动、停止两条信号总线和电源(5V 与地)接 4 只脚，剩下两只空着不用。接插件的电源脚之间，带有吸收噪声

用的电容(47μF,0.01μF)。

与电动机及电动机电源的连接

用螺钉固定式接插件连接电动机的电缆端子。电动机电源用 12V 专用电源,也用螺钉固定式接插件连接。

控制电路

如图 1 所示,口连接管脚的第 0 位管脚(启动、停止信号)与第 1 位管脚(转向信号)接到晶体管 Tr_5、Tr_6 的基极。NOT 电路 N 是为了当转向信号为"0"时,将"1"信号输入到 NAND 电路 NA1。

用于控制的晶体管 Tr_5、Tr_6 与 12V 电源连接,要从接插件端取信号,其原因是:

- 在 NAND 电路的输出为 0 时,要使控制用晶体管导通。
- 由于驱动电动机用的晶体管 Tr_5、Tr_6 与斯密特触发器连接,所以电动机的驱动电压要用 12V,则基极电压也要用 12V。

驱动电路

驱动电路如图 1 所示,将 4 个驱动用晶体管 Tr_1、Tr_2、Tr_3、Tr_4、电动机的接线端子,电源端子连接起来。

Ⅱ 晶体管的选择

驱动用晶体管:电动机的额定电流是 0.5A,则考虑了启动电流时的最大集电极电流 I_c 应当在 3A 以上。

若假定基极电流 I_b 为 0.002A,直流电流放大系数 h_{FE} 可按下式计算:

$$h_{FE}=\frac{I_c}{I_b}=\frac{3}{0.002}=1\ 500$$

因此选用最大集电极电流为 7A,直流电流放大系数为 2 000 的驱动用晶体管 2SD633。

控制用晶体管:最大集电极电流 I_c 是驱动用晶体管基极电流的两倍,即为 0.004A,控制用晶体管的基极电流 I_b,考虑一个与非门电路时的输出电流为 0.000 4A,则直流电流放大系数 h_{FE} 可按下式计算:

$$h_{FE}=\frac{I_c}{I_b}=\frac{0.004}{0.0004}=10$$

可选用最大集电极电流为0.15A，直流电流放大系数h_{FE}为80以上的晶体管2SC1815。

选择理想的晶体管需要一定的经验，可以一边反复试验，一边选用。

运行

停止

若往第0位中输入停止信号0，则两个与非电路的输出为1，两个控制用晶体管导通，全部驱动用晶体管阻断，电动机中没有电流流过。

正转启动

如若第0位中的启动信号为1，在第1位中输入正转的信号1，则与非电路NA2的输出信号为0，使驱动用晶体管Tr_2与Tr_3导通，电动机正转。

反转启动

若第0位中的启动信号为1，在第1位中输入反转的信号0，则非门电路N端为1，并输入与非电路NA1中，其输出变成0，使驱动用晶体管Tr_1与Tr_4导通，电动机反转。

练习题

1. 在接插件电源端子之间为什么要接电容？
2. 非门电路有什么作用？
3. 用控制晶体管Tr_5、Tr_6的目的是什么？
4. 为什么要在直流电动机驱动电路中使用4个晶体管？

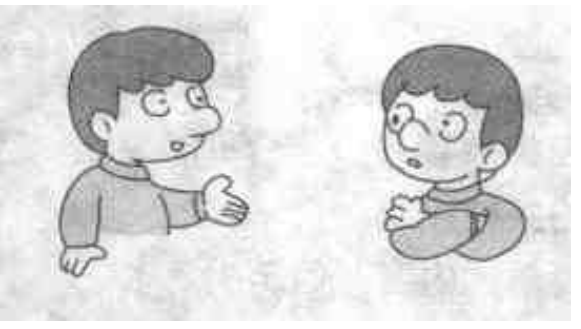


解答

1. 为吸收电源端子前的噪声，防止它串入计算机或接口中。
2. 起到将信号反相的作用。
3. 信号变换与电压变换。
4. 用于改变电流方向，从而改变电动机的转向。

挑战是题

① 8255 是 8 位串行，还是 8 位并行数据传输用的接口？
② 8255 有几个输入输出口？
③ 8255 有几种用法？
④ PC 9801 个人计算机的 16 位数据总线，将低 8 位作为数据的场合，口的地址是偶数还是奇数？
⑤ A、B、C 口是偶数地址时，若用二进制数来表示，应用地址总线的哪几位来改变这几个口的地址？
⑥ 8255 的 CS 管脚有什么作用？
⑦ 为什么数据管脚要通过电阻后再连接到电源上？
⑧ 要让发光二极管点亮所需的电压大约是几伏？是直流还是交流？
⑨ 为什么发光二极管亮灯电路中要用反相器(NOT)电路 IC？
⑩ 向输入输出口输出数据的命令，用 C 语言如何表达？
⑪ 数据的输入命令在 C 语言中是如何表达的？
⑫ 控制字(CW)是什么意思？
⑬ 什么是抖动？
⑭ 斯密特电路是什么电路？
⑮ 电磁继电器电路中，与电磁线圈反向并联的二极管有何作用？
⑯ 小型直流电动机的接口电路中，为什么要用 4 个晶体管？

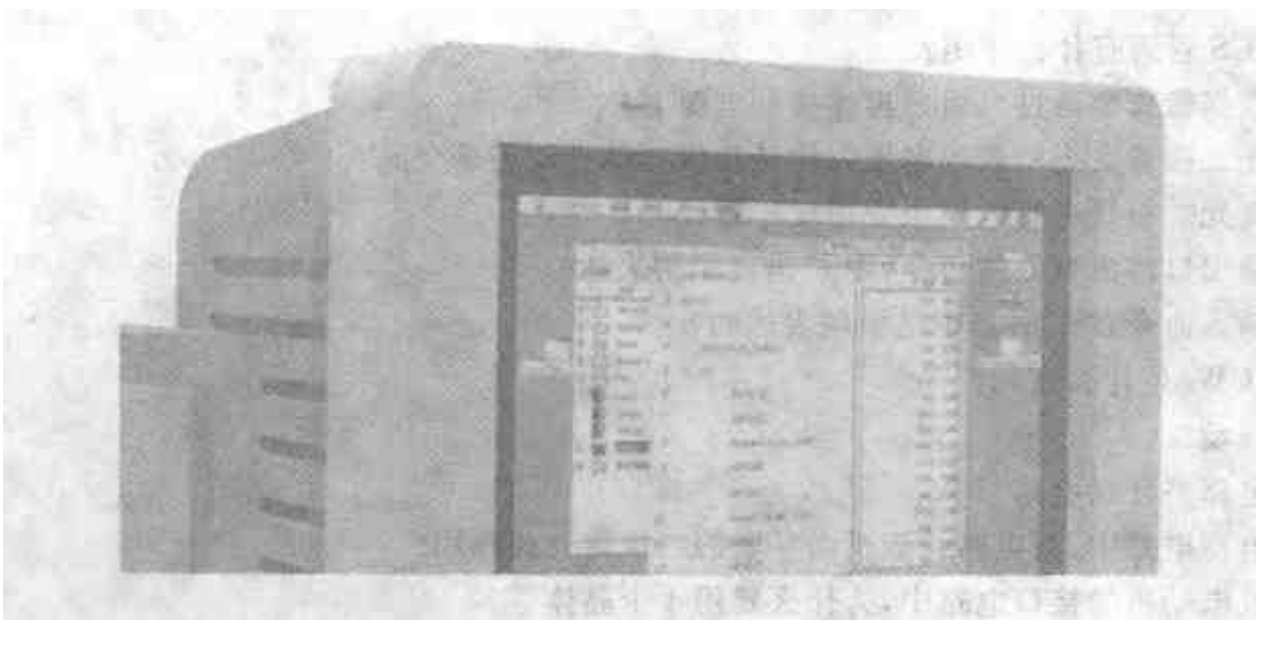


A 答 案

①是 8 位并行数据传输用的接口。 ②A 口、B 口、C 口三个。 ③模式 0，模式 1 与模式 2 三种。④偶数。 ⑤第 1 位与第 2 位。 ⑥CS 是片选管脚。 ⑦为了限制数据信号的电流，称为上拉电阻。 ⑧约 2V 直流。 ⑨为了用流入 IC 内的电流使发光二极管点亮，将输入信号反相，在输出端子上成为“0”。 ⑩outp()：快捷 C(quick)。 ⑪inp()：快捷 C。 ⑫用于8255 接口使用模式设定或表示 C 口各位的设置、复位的 8 位数据。 ⑬自动复位型开关的按动或松开时产生短时间通、断的反复振动波形。 ⑭将有电压状态的输入电压信号分为“1”信号与“0”信号的输出电路。 ⑮为了吸收在切断电磁继电器线圈的电流时产生的反电势。 ⑯为控制电动机的转向，以两个晶体管为一组，任一组导通时，就能改变流入电动机电流的方向。

第7章

简单的控制程序



用C语言编写的简单控制程序

如果在扩展槽中连接了输入输出装置与接口，就可以编制程序，试一下进行输入计算机的各种控制。

“从简单的内容开始努力”。

实际上，与BASIC语言相比，C语言难掌握。定义变量的类型、函数的设计方法等，对初学者来讲是难以理解的内容。但是通过在简单程序中变更条件来反复执行，掌握要领的话就可逐渐理解了。将基本的控制程序写成函数的形式，就能将这些函数编制成复杂的组合程序。

如果是因为困难或复杂对其不进行研究，到时候执行起来困难会更多。所以，首先从使用开关输入电路与发光二极管亮灯电路着手，反复编制并执行各种程序，来掌握要领。

比如，在制作电动机的控制程序时，我们不考虑转速或转向的瞬时变化，而只设计计算按钮开关次数的程序，就可用程序来计转速。

在实践中，输入电路也只用一个开关即可简单地实现。在使转速变化的程序中，若也用发生二极管的亮、灭程序来设计的话，实施也简单，可用人的感官来检验。

另外，实际使用的C语言程序是不带行号的，本书为了说明方便，还是附加了行号。

1 在控制中使用C语言的基本知识



简单介绍一下在控制中常用的C语言的基本知识。

I 数　据

常　量

常量有整数、实数、字符、字符串等。

① 整数：有十进制数、十六进制数等。

• 十进制数，就是以其数值本身来表示。

例如：128、1、8、－32、－153。

• 十六进制数，要在前面带 0x 来表示。

例如：0x10，0xff，0xa2

因在控制中常用十六进制数，请不要忘记在前面带 0x。

② 变量

• 变量名用()中 8 个以内的英文字母、数字来表示，头一个字不能用数字。我想把变量连贯地来看，应当以一定的规律命名，以便于理解为好。

• 在控制中，常用无符号的整型变量，声明例子：unsigned ind x;

• 因＋、－符号占用了 1 位最高位作为符号位，所以数据就少了 1 位，输入输出数据等是不带符号的。

• int 型(整型)变量是计算机处理单元(MPU)能最高效处理的带符号的整数，在用个人计算机的过程中，必须确认(声明)。

Ⅱ　输入输出命令

口输出

outp(地址、数据)

(　) 内的输出目标口地址与输出数据用十进制数或十六进制数标记。

例:outp(0xd6, 0x80)是将数据$(0x80)_{16}$输出到地址号为$(D6)_{16}$的 CW 寄存器中。

例:outp(0xd2,255)是将数据 255 输出到地址号为$(D2)_{16}$的口中。

口输入

x=inp(地址)

(　) 内为输入口地址,表示将输入数据代入变量 x 中。变量名可以任意规定。

例:nyuryok=inp(0xd0);表示将地址号为$(D0)_{16}$的口中的输入数据代入变量 nyuryok 中。

Ⅲ　执行的控制

在控制中是根据输入的条件来决定执行内容的,要同样地反复执行多次时,都用条件分支或循环控制。

条件分支

对开关或传感器等的输入数据作出判断后决定执行时,可用 if 语句或 switch 语句等。结合它们的特点来用。

① if 语句

if () {}

if () {} else {}

if () {} else if () {} else {}

在这些可用的方法中,()内写条件表达式,{}内写执行语句,有多个条件表达式,也能判断执行。

例如:if(条件式 1||条件式 2)

意思是条件式1与条件式2中任一个条件式成立的话。

例如：if(条件式1&&条件式2)

意思是假设条件式1与条件式2都成立。

② switch(开关)语句

switch(表达式)的语句中，表达式所指的值，可用执行语句case来处理，并且只能执行其中一个表达式。

case语句中若不插入break语句，则继续执行随后的语句，若插入break语句，则switch语句结束，执行switch语句之后的执行语句。

循环控制

在条件表达式成立之前，对执行语句反复执行的场合，使用无条件循环判断语句。While语句、do~while语句等。

① 无条件循环

• 若省略for语句的三个条件表达式，则成为无条件循环语句。

```
for (;;) {}
```

• while语句的条件表达式中，若使用除0之外的数字，则循环就无限制地延续，程序不会停止执行。

```
while (1) {}
```

② 从循环圈中跳出

用判断语句(if语句或switch语句)与break语句或goto语句均可跳出循环圈。

break语句跳出循环圈是执行后面的语句，用goto语句跳出则执行指定标记处的语句。

在程序结构中，还是少用goto语句为妙。

③ 在循环执行过程中间，进行中断处理，然后再继续循环时，要通过判断语句执行continue语句的话，则不再执行以后的语句，而进入for语句的最早的执行语句。

④ while语句与do~while语句

这两个语句都是对执行语句的条件是否成立作出判断的，它们作出判断后的执行有所区别，但在条件表达式成立期间，反复地执行循环这一点是相同的。

由于while(条件表达式)中只能设置一个条件表达式，所以不能进行IF语句中那种复杂的判断。但是IF语句中的循环条件判断必须借用其他语句的功能来实现。

*　　*　　*

在某种程度上，想要编制使用开关的输入电路与发光二极管亮灯电路的程序，还是有必要用实际的装置来执行所用的控制程序。在实际的控制中，会出现理论上考虑不到的问题。

编制的程序不仅需要程序的知识，还牵涉到必要的电气、电子的基础知识和机械等方面的知识，也有用程序控制解决不了的问题，而必须用硬件的手段来解决。

练习题

1. 程序中十进制数与十六进制数在表示上有什么不同？
2. 命名变量名称有什么规定？
3. while 语句与 do～while 语句有什么区别？
4. 有什么办法可从循环圈中跳出？



解答

1. 十进制数就用数值本身表示，十六进制数要在前面附加 0x 表示。
2. 用 8 位以内的字母、数字写入()内，头一个字不能用数字。
3. while()是通过条件判断来执行，而 do～while()则是通过执行来作出条件判断。
4. 使用判断语句、break 语句或 goto 语句等。

2 8位发光二极管的亮、熄程序

将8位发光二极管接入8255输入输出接口的B口中，观察各种使发光二极管亮、熄的情况(参见图1)。

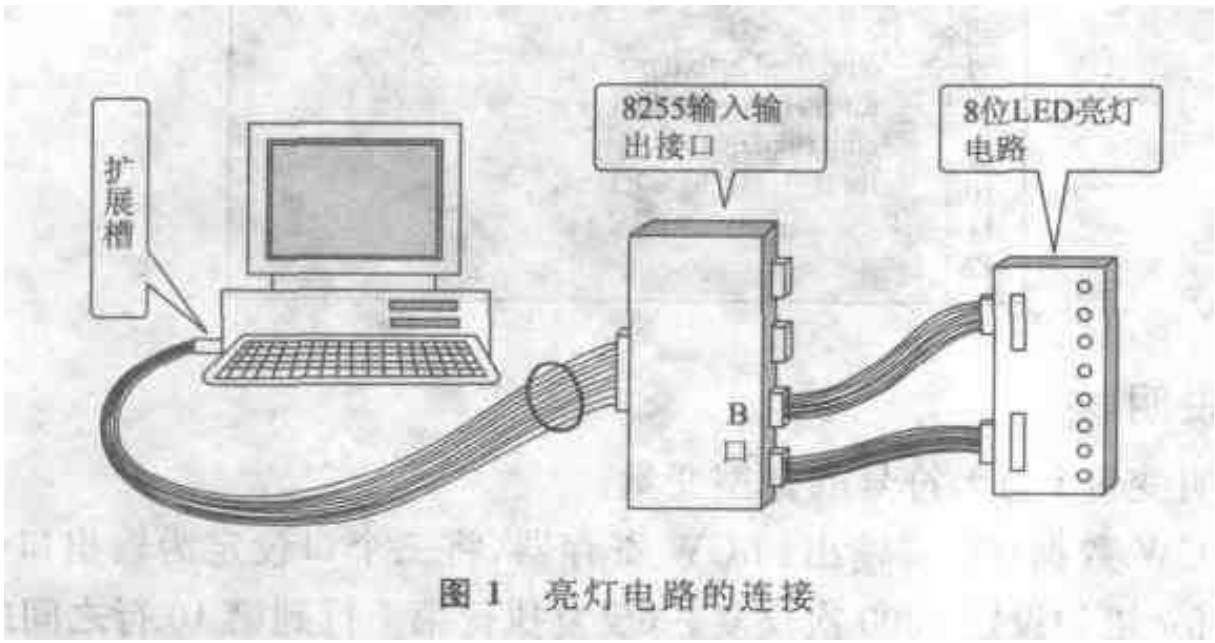


图1　亮灯电路的连接

Ⅰ 全部灯亮、熄

编制使全部灯反复亮、熄1000次的简单程序(参见图2)。



图2　亮灯模式

数　据

CW 寄存器地址：$(D6)_{16}$。

B 口地址：$(D2)_{16}$。

CW 数据：$(80)_{16}$。

8 位亮灯数据：$(ff)_{16}$。

8 位熄灯数据：$(0)_{16}$。

程　序

```
1  main ()
2  {
3    unsigned int i;
4    outp (0xd 6, 0x80);
5    for (i=1; i ▮ =1000; i++)
6    {
7      outp (0xd 2, 0xff);
8      for (i=1; i<=30000; i++);
9      outp (0xd2,0x0);
10     for (i=1; i<=30000; i++);
11   }
12 }
```

程序说明

3. 声明变量 i 为无符号的整型变量。

4. 将 CW 数据$(80)_{16}$输出到 CW 寄存器，将三个口设定为输出口。

5. 用 for 语句设定 1000 次反复。（反复执行第 7 行到第 10 行之间的命令。）

7. 向 B 口输出全部灯亮的数据$(ff)_{16}$。

8. 为延长亮灯时间，用条件循环语句，反复的次数来延长 B 口输出的保持时间。

9. 向 B 口输入全部灯熄灭的数据$(0)_{16}$。

注意：计算机的处理速度极快，是以 μs 为单位间隔亮、熄灯的，人所能判断的速度不过 10Hz 左右，所以要通过第 8 行与第 10 行的 for 语句来延长亮、熄的时间，才能判断亮、熄。

Ⅱ　单方向亮、熄程序

发光二极管将从第 0 位到第 7 位一个个地依次亮、熄，编制这样反复 1000 次的控制程序。

程　序

```
1  main ()
2  {
3      unsigned int od,n,i,t;
4      outp (0xd6,0x80);
5      for (i=1; i<=1000; i++)
7      {
8        od=0x1;
9        for (n=0; n<=7; n++)
11       {
12           outp (0xd2,od);
13           for (t=1; t<=3000;t++);
14           od<<=1;
15       }
16     }
17 }
```

程序说明

8. 将$(1)_{16}$代入变量 od 中。

9. 设置从第 0 位到第 7 位依次使灯亮熄的 for 语句。(将第 11 行到第 14 行之间的命令,代换数据,反复 8 次。)

Ⅲ　反复往返亮灯

若在 2 的程序第 15 行与第 16 行之间追加从下 6 位起向第 1 位以相反的顺序亮灯的程序,则能进行往返亮熄。

追加程序(返回的亮、熄)

```
1  for (n=1; n<=5;n++)
2   {
3     od>>=1;
4     outp (0xd2,od);
5     for (t=1; t<=3000;t++);
6  }
```

练习题

1. 无符号整型量与有符号整型量有什么不同?
2. 无条件循环的 for 语句是干什么用的?
3. od<<=1 与 od>>=1 有什么区别?

解答

1. 用+、-符号要用到最高位,作为数来用要少一位。无符号量则可以将所有的位都作为数来用。
2. 为了延长前面输出的数据在口中保持的时间。
3. od<<=1 是将 od 的位数据往左移 -位的命令,而 od>>=1 是将 od 的位数据往右移 --位的命令。

3 利用开关的控制

8255接口的A口上连接8位开关的输入电路，B口上连接发光二极管电路，C口上连接电磁继电器的驱动电路，来试一下用开关输入的控制(参见图1)。

Ⅰ 用开关的控制

功　能

当 A 口的第 0 位的开关 ON(“1”,接通)时,B 口的 8 位全部发光二极管均熄灭,当第 7 位 ON(接通)时,程序结束。

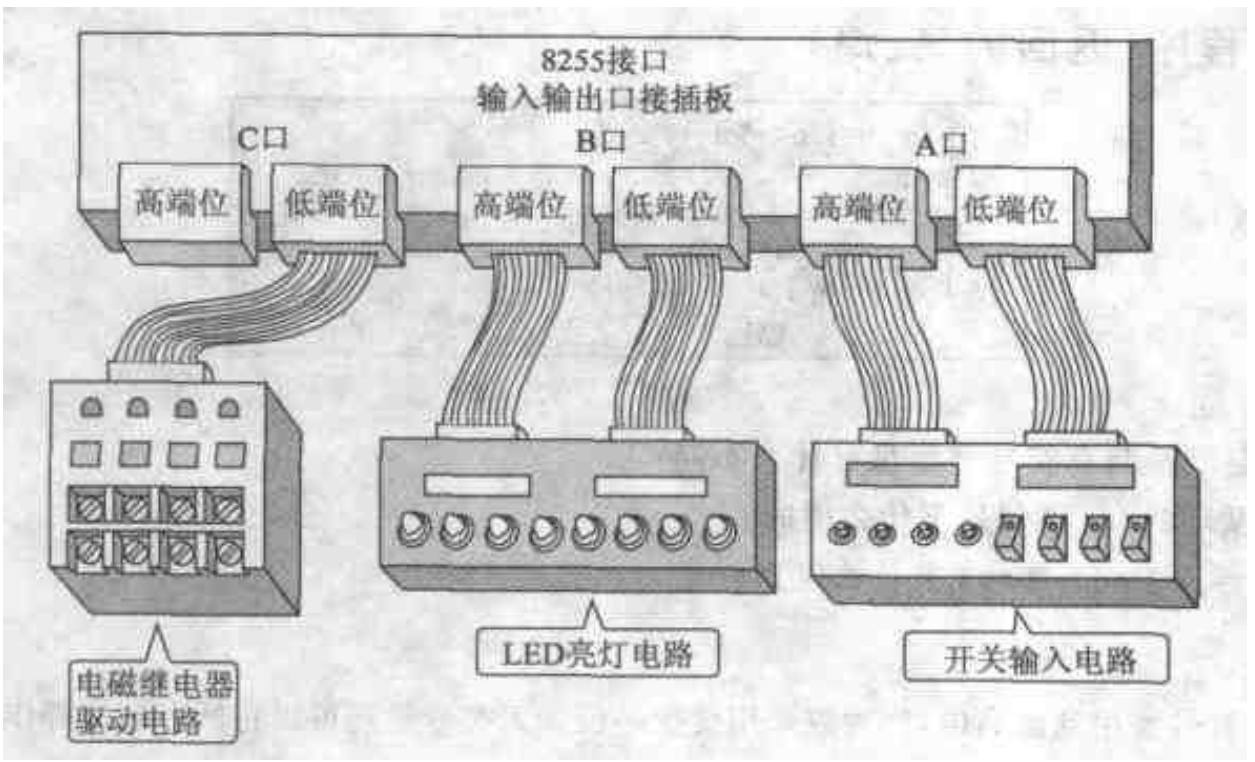


图 1　连接结构

程　序

```
1  main ()
2  {
3      unsigned int sw;
4      outp (0xd 6, 0x 90);
5      do {
6          sw=inp (0xd0);
7          if (sw = =1)
8          {
9              outp (0xd2,0xff);
10         }
11         else
12         {
13             outp (0xd2,0x0);
14         }
15     }while(sw!=128);
16 }
```

程序说明

5～15. do～while 语句，在按第 7 位的开关之前，反复执行第 6 行到第 14 行之间的命令。

6. 将 A 口的数据输入变量 SW 中。

7～14. 用 if 语句，当输入数据为 1 时，将$(ff)_{16}$输出到 B 口中，全部发光二极管点亮，其余情况下将$(0)_{16}$输出到 B 口，使发光二极管熄灭。

Ⅱ　用多个开关控制

假定按表 1 所示对 A 口的开关进行操作，使 C 口的电磁继电器动作。

表 1

A 口的开关	C 口的电磁继电器
第 4 位 ON(接通)	第 0 位工作
第 5 位 ON(接通)	第 1 位工作
第 6 位 ON(接通)	第 0 位与第 1 位工作
第 7 位 ON(接通)	所有位均停止工作

若按第 0 位开关的话，程序结束。

程　序

```
1  main()
2  {
3      unsigned int sw;
4      outp (0xd6,0x90);
5      do {
6          sw=inp (0xd0);
7          switch (sw)
8          {
9            case 0x10;
10             outp (0xd2,0x1);
11             break;
12           case 0x20;
13             outp (0xd2,0x2);
14             break;
15           case 0x40;
16             outp (0xd2,0x3);
17             break;
18           case 0x80;
19             outp (0xd2,0x0);
20         }
21       } while (sw!=1)
22 }
```

程序说明

5～21. do-while 语句的条件(不接通第 0 位开关的情况下)成立，则以第 6 行到第 20 行循环，若条件不成立，则程序结束。

8～20. 开关语句，根据各 case 常数表达式的条件，执行各自的输出。例如，在 case 0x10;的条件下，变量 SW 的值是$(10)_{16}$时，执行随后命令。

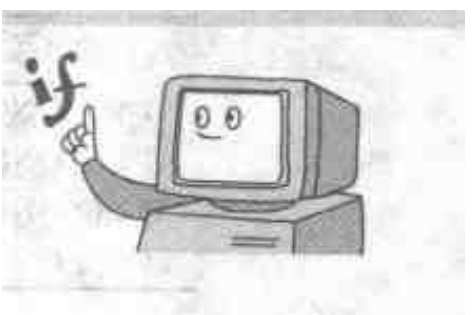
break 语句是使 switch 语句结束。

★ 对开关等的输入数据进行判断处理时，使用 if 语句或 switch 语句。判断语句多的场合，用 switch 语句则更方便。

用 if 语句时，需要进行许多的条件检查。

在最后的 case 语句中，只所以没有 break 语句，是因为不带 break 语句，switch 也会结束的缘故。

练习题



1. if 语句中的()的条件中，将＝＝写成＝是错的吗?
2. if 语句中的 else 是什么意思?
3. switch()语句的 case 是什么意思?
4. break 是作什么处理的?

解答

1. 是错的。＝是代入的意思，不是等号。
2. if 语句中()的条件不能成立时，执行 else 后面的程序。
3. switch 语句的变量用 case 表示数据的场合，执行后面的处理。
4. 中断现在为止的处理，执行下一个处理。

4 利用传感器的控制

编制一个程序，用温度传感器检测温度，当室内温度到达上限时启动鼓风机，使温度下降，当温度到达设定温度下限时，使鼓风机停止。

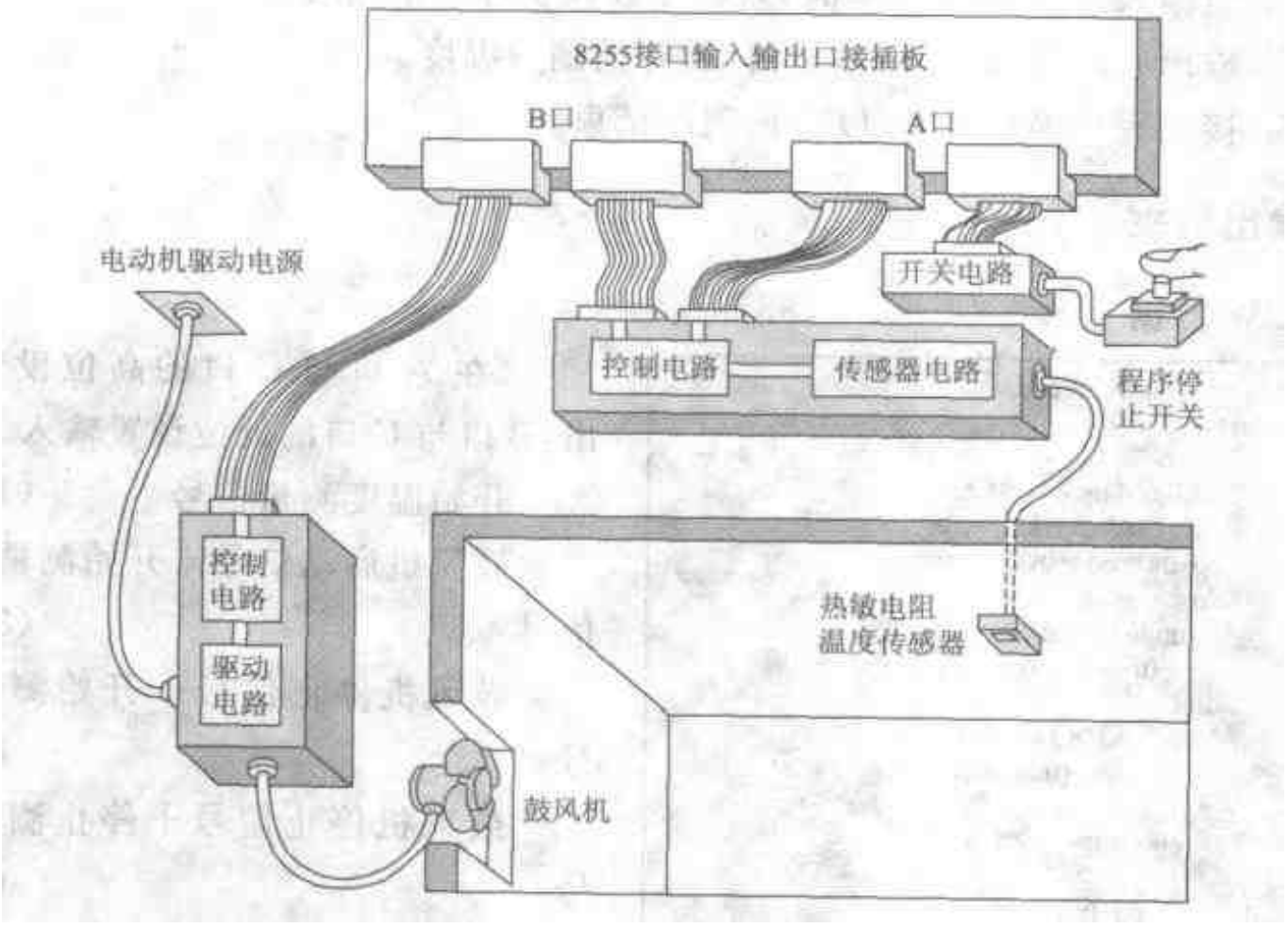


图 1 温度控制连接

1 连接与功能

连 接

结束程序的开关输入信号，接在 A 口的第 0 位上，热敏电阻温度传感器的

30℃以下的信号，接在 A 口的第 4 位上，35℃以上的信号，接在 A 口的第 5 位上，分别作输入信号。

开始检测的输出信号接在 B 口的第 0 位上，鼓风机启、停信号接在 B 口的第 1 位上。

动　作

1. 将开始测量的信号输出到传感器，输入温度数据后输出停止测量的信号。

2. 因测量温度与处理信号要花一定的时间，所以经过一定时间后，输入测量结果。

3. 当输入 35℃以上的信号时，就输出鼓风机启动信号。

4. 当输入 30℃以下信号时，就输出鼓风机停止的信号。

5. 输出鼓风机的启、停信号后，就开始测量温度。

6. 接通程序停止开关(ON)使程序结束。

输出数据

CW 数据　　$(98)_{16}$

（在 A 口与 C 口的高位设置输出，B 口与 C 口的低位设置输入。）

开始温度测量信号　　$(1)_{16}$

鼓风机启动信号＋开始测量信号　　$(3)_{16}$

鼓风机停止信号＋开始测量信号　　$(1)_{16}$

鼓风机停止信号＋停止测量信号　　$(0)_{16}$

输入数据

35℃以上信号　　$(20)_{16}$

30℃以下信号　　$(10)_{16}$

程序结束信号　　$(1)_{16}$

程　序

见右侧程序。

```
int ondo (int od,int odd,int tt);
main ()
 {
      unsigned int od,odd,tt,st;
      outp (0xd6,0x98);
      od=0x1; odd=0x0;
      do {
          ondo (od,odd,tt);
          if (tt==0x20)
          {
              od 0x3;odd=0x2;
              outp (0xd2,od);
          }
          else if(tt==0x 10)
          {
              od=0x1; odd=0x0;
              outp (0xd2, 0x0);
          }
          st=inp (0xd0) & 0x1;
     } while (st!=0x1);;
 }
ondo (int 0d,int 0dd,int tt)
 {
     unsigned int i,t,;
     outp (0xd2,od);
     for (i=1;i<=1000;i++);
     t=inp (0xd0);
     tt=t& 0xf0;
     outp (0xd2, odd);
 }
```

程序说明

1. 声明函数 ondo()的原型。

6. 设置 od、odd 的初值。

7～20. do-while 语句，当开关输入为 0x1 时结束程序。否则反复地执行语句内容。

8. 调用函数 0ndo()，进行温度测量。

9～18. if 语句，当 ondo()中的变量 tt 等于 0x20 时(测量的温度在 35℃以上)，将“3”输出到 B 口的第 0 位上，tt 等于 0x10 时，(测量温度在 30℃以下)，将“1”输出到 B 口的第 0 位上。

19～27. 是函数 ondo()的定义。

练习题



1. 热敏电阻是测量什么的传感器？
2. 叙述输出开始测量信号时，不能立即输入温度数据的理由。
3. 为什么要交接输出数据？
4. 输入传感器数据时，应注意什么问题？

解答

1. 是温度传感器。
2. 测量、处理输入计算机的数据要花一些时间。
3. 用函数 ondo()输出开始与停止测量数据时，由于鼓风机的启动时与停止时数据不同的缘故。
4. 是输入延时。

5 汽缸的控制

工厂里发出机械的嘘嘘声，就是汽缸等利用空气压力工作的机械发出的声音。汽缸是用于直线运动的执行机构。这里试编一个汽缸的简单控制程序(参见图1)。

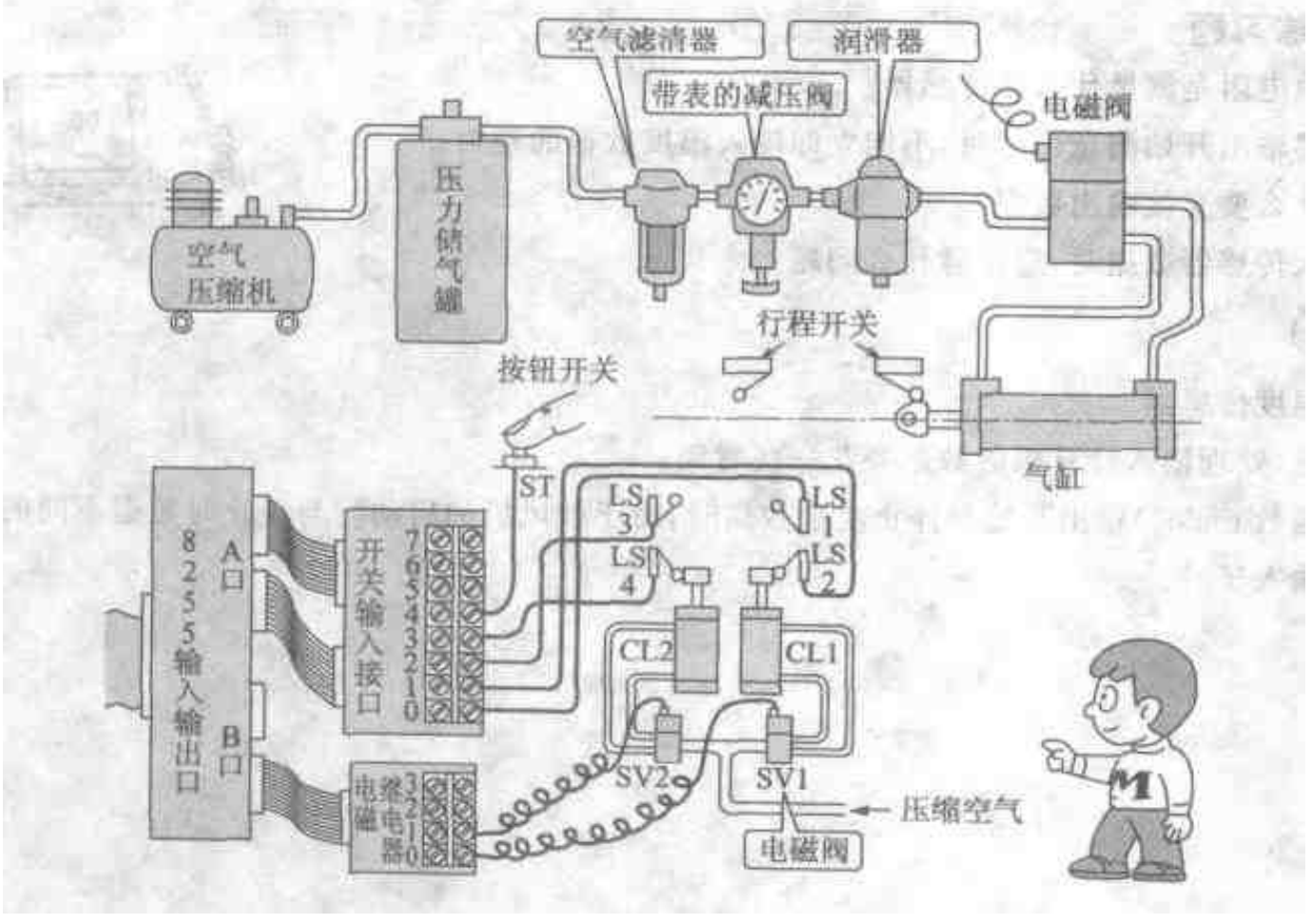


图 1　汽缸工作系统

I 汽缸系统

汽　缸

汽缸是在空心圆筒(缸体)中装一个圆盘(活塞)，用压缩空气的能量作直线

运动的机构。压缩空气通过活塞移动方向后端的进气口送入，并将活塞移动方向前端的空气排到外面去。

空气压缩系统

用空气压缩机产生的压缩空气，经过空气滤清器，滤去空气中的灰尘和水分，通过带表的减压阀来调节工作压力，用喷雾润滑器等向压缩空气喷洒润滑剂，来向空压机提供润滑油，然后再将这压缩空气送入汽缸，这就组成空气压缩系统。在耗气量大的情况下，要附加压力储气罐以防止压力降低。

电磁阀

向汽缸提供的压缩空气，是通过螺线管驱动的电磁阀来控制的。

一般的电磁气阀，大多是单端螺线管式，根据是否有电流流过螺线管，对电磁阀进行通断控制。

位置传感器

用行程开关检测活塞的位置，通过开关用的接口将信号输入计算机。

电磁阀驱动电路

用来自计算机的信号，控制驱动电路的电磁继电器，通过继电器接点来接通、切断电磁阀的交流电源。

与计算机的连接

通过开关用的接口，将 4 个行程开关与按钮开关连接到 8255 输入输出接口的 A 口上。B 口上连接电磁继电器的驱动电路，将电磁阀连接到继电器的接点上。

Ⅱ 控 制

动 作

① 若按下按钮开关，电磁阀 SV1 工作，汽缸 CL1 的活塞伸出，压住行程开关 LS1。

② 若压住 LS1，电磁阀 SV2 工作，汽缸 CL2 的活塞伸出，压住行程开关 LS3。

③ 若压住 LS3，则电磁阀 SV1 停止工作，汽缸 CL1 的活塞缩回，压住行程开关 LS2。

④ 若压住 LS2，则电磁阀 SV2 停止工作，汽缸 CL2 的活塞缩回，压住行程开关 LS4。

程　序

```
int dousa (int id,int od,int iid)
main ()
{
    outp (0xd6,0x90);
    do{
        id=0x1a;
        od=0x1;
        dousa (id,od,iid);
        id=0x9;
        od=0x3;
        dousa (id, od,iid);
        id=0x5;
        od=0x2;
        dousa (id,od,iid);
        id=0x6;
        od=0x0;
        dousa (id,od,iid);
    } while 1;
}
dousa (int id,int od,int iid);
{
    unsigned int id,od,iid;
    do {
      iid=inp (0xd0);
    } while (iid!=id);
    outp (0xd2, od);
}
```

练习题

1. 汽缸的活塞动作时，压缩空气是通过哪个进气口进入的？
2. 要改变汽缸压缩空气进气口气流方向，使用什么东西来控制？
3. 要将电磁阀与计算机连接时，通过什么输入输出接口？
4. 用什么检测活塞的位置？

解答

1. 从动作方向后端的进气口进入。
2. 用电磁阀(换向阀)。
3. 电磁继电器的驱动电路。
4. 是行程开关等。

6 直流电动机的控制

试编写一个控制小型直流电动机的转速与方向控制程序，此电动机带减速机，在输出轴上装有圆盘。转速是通过该圆盘上的一个孔与一体式透光型传感器来检测的(参见图1)。

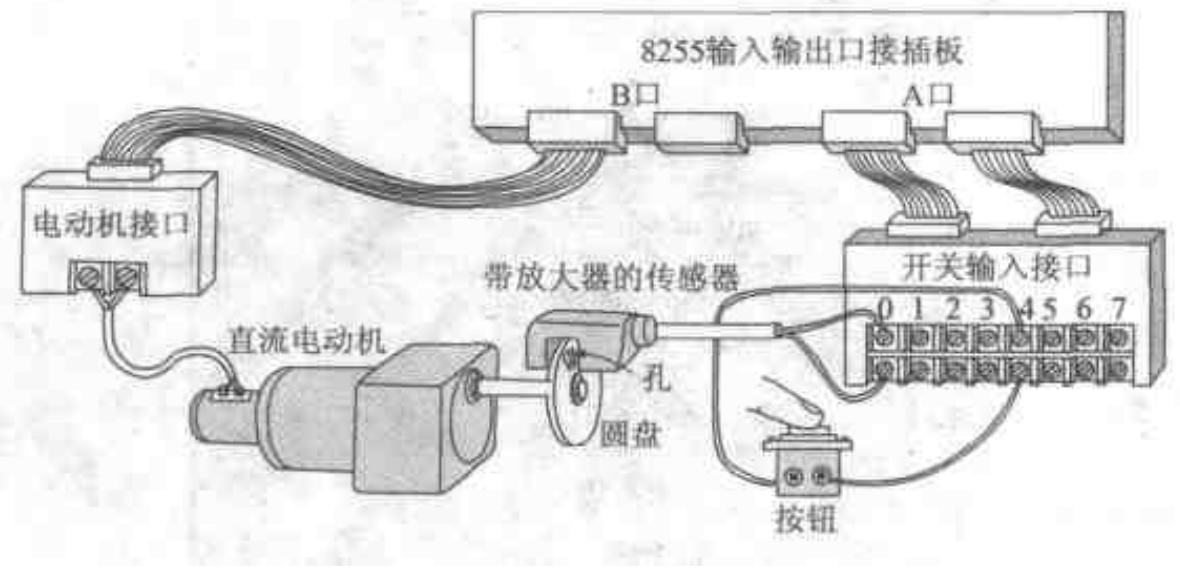


图1　直流电动机控制系统

连　接

在8255输入输出接口的B口上，接入小型直流电动机的驱动电路。通过开关输入接口分别在A口的第0位上接光传感器，在第4位上连接按钮开关。

动　作

① 若按下第4位的按钮开关，电动机开始正转。

② 用光传感器检测转速到100转为止，反转(圆盘上的孔若透光，则传感器输入“1”的信号，无孔之处，均输入“0”信号)。

③ 若反转到200转，就停止。

输入数据　　开关 ON(通)：0x10

输出数据　　正转 ON(通)：0x1

　　反转 ON(通)：0x3

程　序

```
1  int kaiten(int n,int od);
2  main ()
3  {
4          unsigned int st,stp;
5          outp (0xd6,0x90);
6          do{
7              st=inp(0xd0) & 0x10;
8          }while(st!=0x10);
9          n=100;od=0x1;
10         kaiten(n,od);
11         n=200;od=0x3;
12         kaiten(n,od);
13 }
14 kaiten(int n,int od)
15 {
16         unsigned int n,nn,nh,ls,h,od;
17         h=0x1;nn=0;
18         nh=n*2;
19         outp(0xd2,od);
20         do{
21             ls=inp(0xd0) & 0x1;
22             if(ls==h)
23             {
24                 nn=nn+1;
25                 h=~h;
26                 h=h & 0x1;
27             }
28     }while(nn!=nh);
29     outp(0xd2,0x0);
30 }
```

程序说明

1. 定义函数 kaiten()。规定函数的名称，声明变量的顺序与类型。

6～8. 在启动开关 ON(接通)之前，反复执行 do-while 语句。

7. 此语句用于输入 A 口的数据，并将该数据与 0x10 进行逻辑与(AND)运算，再将第 4 位的数据代入变量 st 中。

变量 st 若满足 8 的条件语句(不等于 0x10)则在内部进行反复循环，若不满足条件，则结束循环。

9. 将转数(100)代入变量 n，将正转启动输出数据(0x1)代入变量 od 中。

10. 调用检测转数累计值的函数 kaiten(n,od)。

11. 将转数(200)代入变量 n，将反转启动输出数据(0x3)代入变量 od。

12. 调用检测转数累计函数 kaiten(n,od)。

13. main()函数结束。

14～30. 函数 kaiten(n,od)的程序。

17. 将初始值代入变量 h,nn 中。

18. 22～27. 的 if 语句,是为了累计输入数据 ls 变化后的转数,将转数的变量 n 扩大为 2 倍。

19. 将启动旋转信号从 B 口输出。

20～28. 累计转数,用 do-while 语句判断是否是到了指定转数。

21. 将来自 A 口的输入数据与 0x1 进行按位的逻辑与(AND)运算,并代入提取转数的传感器的数据变量 ls 中。

22～27. 用 if 语句累计转数传感器的输入数据 ls 的变化次数。

22. 输入数据 ls 与 h 相等的情况下。

21. 变量 nn 加 1,nn=nn+1 是累加式。(也可以写成 nn++1)。

25. 为了看到输入数据的变化,对比较数据 h 求反。

~是按位的逻辑非运算符。

(0000000000000001)→(1111111111111110)。

26. 因为是全部的按位求反,用 0x1 进行按位逻辑与运算,就取得了第 0 位。

28. 判断是否到了指定的转数,若转数变量的累计值 nn 不等于指定的转数变化值时,则在内部反复循环,若相等就停止循环,执行下一个语句。

29. 将停止的信号输出到 B 口,结束函数 kaiten(n,od),返回主程序 main()。

★ 由于转速的检测是通过光传感器输入信号的变化时的转数累计得到的,所以合计的转速是 1/2 转数。

练习题

1. 想提取特定位的数据时,为什么要进行按位的逻辑与(AND)运算?
2. if 语句的关系运算符! =是什么意思?
3. 第 28 行的}while(nn! =nh)是什么意思?
4. ~h 是什么意思?

解答

1. 按位逻辑与(AND)运算,"与"0 则全部都是 0,"与"1 则只留下原先的 1 位。
2. 不等于;不等于时。
3. 当 nn 不等于 nh,则 do{之间的命令。
4. 按位的逻辑非,进行位的反演。

7 步进电动机的控制

用驱动控制电路IC，通过步进电机驱动电路，对步进电动机进行控制，其控制信号是脉冲信号与转向信号(参见图1)。

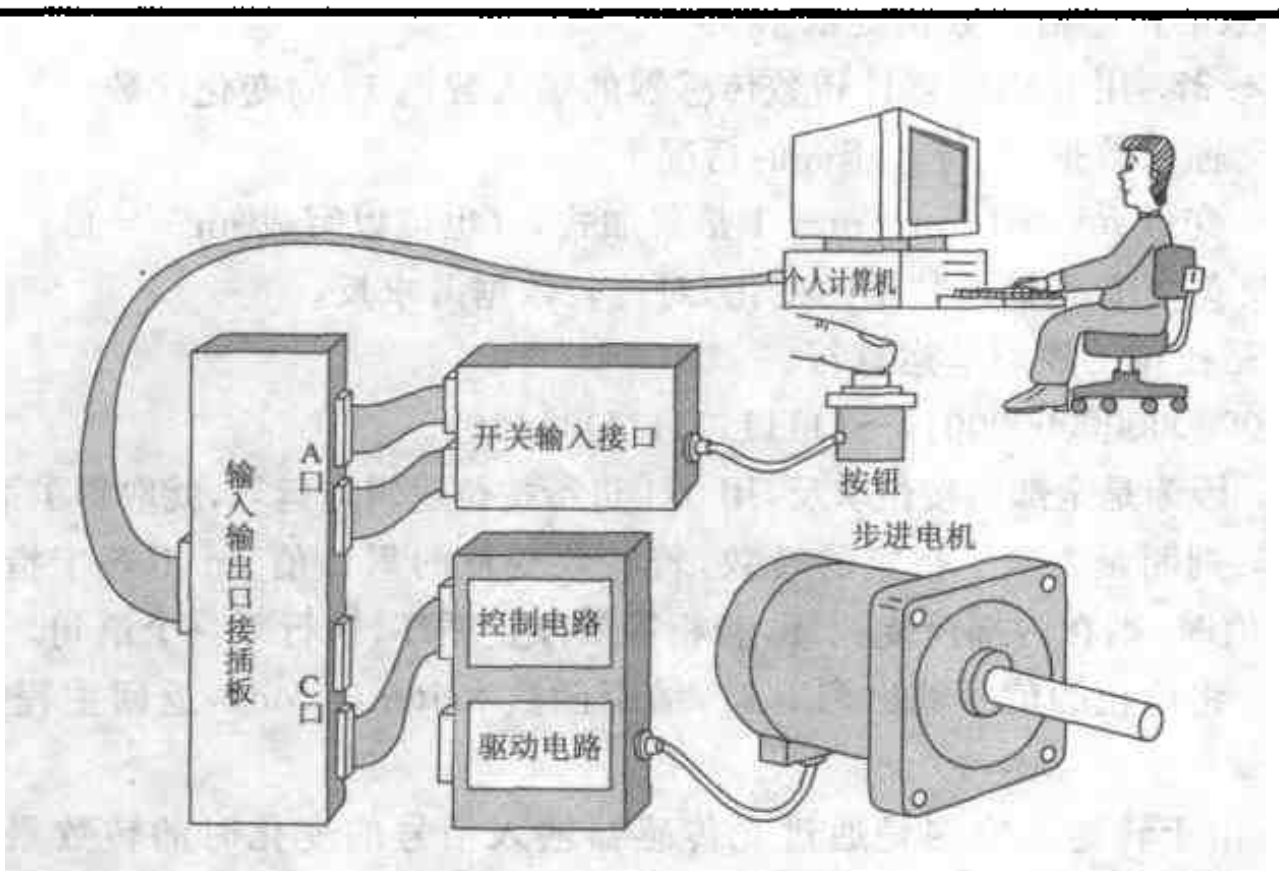


图1　步进电机控制系统

I 信号与连接

脉冲信号

脉冲信号是用0V与5V电压信号互相交替的连续信号，输入到步进电机的驱动电路中，使所连接的电动机旋转。一个脉冲的转角是1.8°。以200个脉冲转一圈。转速根据脉冲的周期来决定。若停止脉冲信号的输入，电动机就停止旋转。

通过键盘，可再次执行转数、正转、反转、启动的输入。

连　接

步进电动机的驱动电路，接在 C 口上，脉冲信号从第 0 位输出，转向信号从第 1 位输出。

II 控制程序

动 作

① 键入转向与转数(脉冲数)。

② 由键盘发出启动信号。

③ 到达指定的旋转量，就停止旋转，返回到等待输入状态。

④ 通过键盘再执行程序，停止输入。

程 序

见左侧程序

程序说明

6～32. 是控制程序反复循环，还是结束的 do-while 语句。

7～8. 用 printf()语句在屏上显示键入的内容，用 scanf()语句将键入转数代入变量 n 中。

9～24. 根据键入所对应的转向，决定转向数据 h_0 值，在没有输入的情况下，则重新执行 do-while 语句。

25～28. 在键盘输入启动电动机的指示之前，反复执行 do-while 语句。

29. 调用脉冲输出函数 pulse(n, ho)。

31. 通过键盘的输入来指示再次执行还是结束程序。

```
1  int pulse (int n,int ho);
2  main ( )
3  {
4      unsigned int n,h,ho,id,st;
5      outp (0xd6,0x90);
6      do {
7          printf ("旋转量=");
8          scanf ("%d", &n);
9          do {
10             printf ("方向(cw=1,ccw=0)=");
11             scanf("%d",&h);
12             if (h= =0)
13             {
14                 ho=0x0;
15             }
16             else if (h= =1)
17             }
18                 ho=0x2;
19             }
20             else
21             {
22                 h=2;
23             }
24         } while (h= =2);
25         do {
26           printf ("启动(1)?=");
27           scanf ("%d", &k);
28           } while (k!=1);
29         pulse (n,ho);
30         printf ("再执行一次?(是=1)");
31         scanf ("%d",&r);
32     } while (r= =1)
33 }
34 pulse (int n,int h)
35 {
36     unsigned int i,t,odh,odl;
37     for (i=1;i<=n;i++)
38     {
39         odh=ho+0x1;
40         odl=ho+0x0;
41         outp (0xd2,0dh);
42         for (t=1; t<=100;t++);
43         outp (0xd2,odl);
44         for (t=1; t<=100;t++);
45     }
46 }
```

34～36. 脉冲输出函数 pulse(n,ho)。

37～45. 用 for 语句，在脉冲数达到所指定变量 n 的值之前，反复执行 39～41 语句。

42～44. 用 for 语句，在一定时间内，保持输出。

练习题

1. 步进电动机的转速如何调整？
2. 什么是键入函数？
3. scanf()函数之前的 printf()函数有什么作用？



解答

1. 用脉冲的周期。
2. scanf()。
3. 只用 scanf()函数不能显示任何屏面，难以理解，所以要用 printf 函数在显示屏上显示 scanf 函数的内容。

Q　挑战题

① "unsigned int i;"是什么意思?

② "otpu(0xd6,0x80);"是什么意思?

③ "od<<=;"是什么意思?

④ 编写一个将变量 i 从 1 到 2000 各代入程序一次的程序命令语句。

⑤ 写出变量 r 等于 1 时,将变量 x 的内容输出到 B 口上,若不等于 1 时,则输出到 C 口上的 if 语句。

⑥ 变量的值有几种? 根据值的不同来执行程序,用什么命令?

⑦ 在 switch 语句中,相当于 if 语句中的 else 的内容是什么?

⑧ 用 goto 语句如何表示跳转的目标?

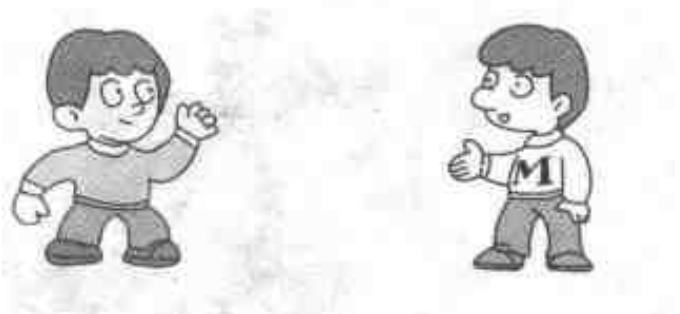
⑨ 要在数据的任一位上,提取二进制数时用什么办法好?

⑩ 汽缸的活塞如何推动?

⑪ 如何控制向汽缸的各进气口切换压缩空气?

⑫ 用光传感器检测电动机转数的基本原理是什么?

⑬ 为了使步进电机旋转,需要什么样的信号?



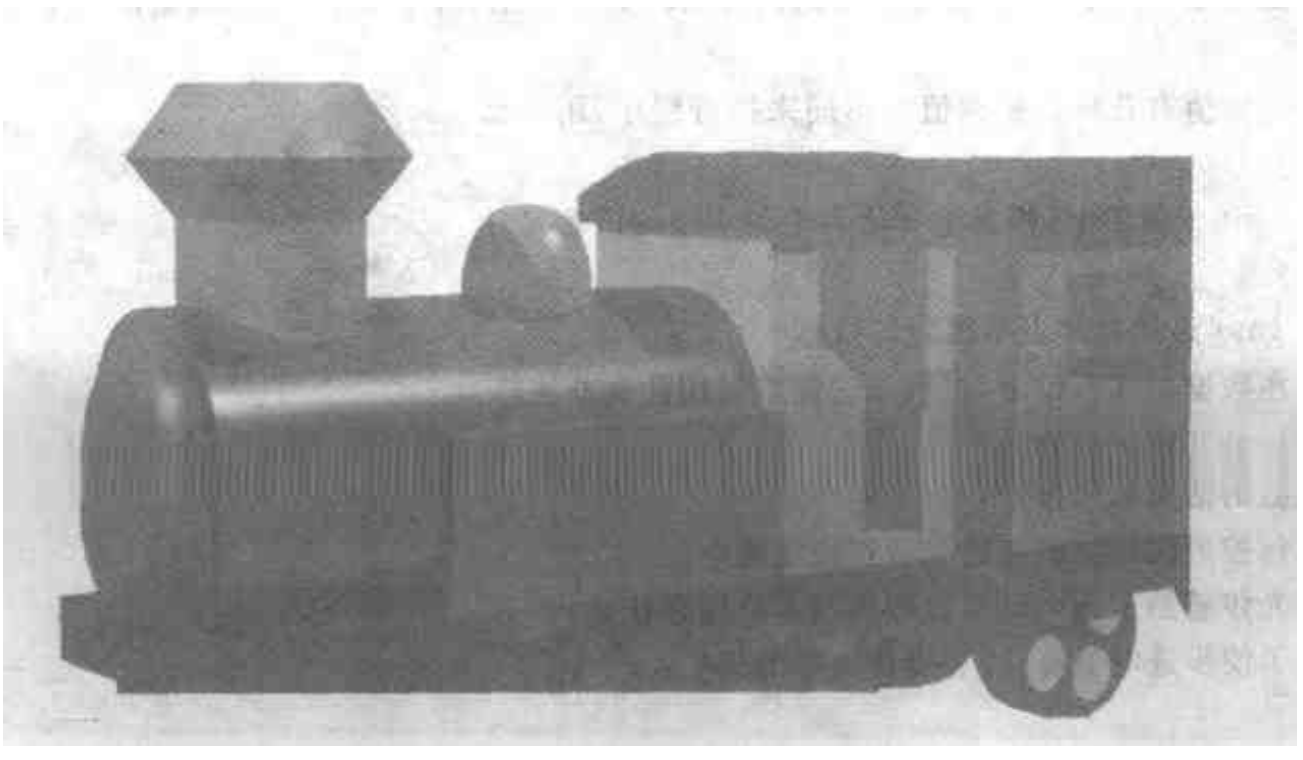
A　答　案

①将变量 i 定义为无符号的整型变量。　②将 CW 变量$(80)_{16}$输出到地址为$(d6)_{16}$的 CW 寄存器中,设定输入输出的使用条件。　③是将变量 od 的内容的二进制数值左移 1 位的命令。　④ for(i=1;i(=200;i++)。　⑤程序为:　⑥用开关语句。　⑦缺省值。　⑧指定的标记处。⑨只要将任意位设置为 1,其他位均为 0 的数据进行按位的 AND(与)逻辑运算。　⑩向动作方后端的进气口输入压缩空气。　⑪电磁阀:用电磁阀进行换向。　⑫通过旋转将光遮挡与透过来检测,其变化数为转数的 1/2。　⑬脉冲信号。

```
if(r==1)
{
    outp(0xd2,x);
}
else
{
    outp(0xd4,x);
}
```

第8章

简单装置的控制



从理论到实践、挑战简单控制

我们已经学过了简单的输入输出装置的控制，在本章将学习更多装置的控制。

照相机、电子表、电子煤气灶、洗衣机、热水器、自动门、电梯、各种自动售货机、数控机床、工业机器人、各种自动机床、铁路系统等，从身边的家用电器到大型的系统，使用计算机控制的东西非常多。由于对这些控制提出方便性、安全性等多种功能的要求，使控制也变得复杂、困难了。

第一，试制几个简单的模型来学习基本的控制。从自动控制的基本方法，以及自动机床常用的顺序控制等入手，对使用气动执行机构顺序控制的机器人进行控制。

第二，试制一个我们经常接触的自动门。因实际的自动门需要特别注意安全，控制很复杂，所以这里只学习基本的控制。

第三，试一下使用步进电机的传动装置的控制。这种传动装置正大量地用于计算机外围设备的执行机构中。

第四，试一下对铁路模型的控制。新干线均由计算机控制，为理解其基本控制方法，使用了铁路模型，但由于篇幅所限，主要学习铁路模型的基本控制。

由于大多数场合均用C语言编制的控制程序，所以这里也用C语言编的控制程序。因控制只考虑基本的内容，所以程序也只考虑基本的内容。我想作为一种学习程序，从编制简单装置的程序开始较好。对控制来说，积累技术诀窍，并在调试过程中反复修正错误，改进设计，也能取得成效。我觉得不断地学习与试验是成功的秘诀。

1 用气动执行机构的传送装置的控制

自动机床所用的执行机构大多数是电动机，但气动执行机构有其自身的特点，也用得相当多。气动操作有以下特征:

① 力量大，变速方便，容易获得高速操作(1000mm/s左右)。

② 由于气体有可压缩性，能吸收冲击，可平滑地操作。

③ 用简单的机构就可以产生直线运动或旋转运动。

④ 可在有爆炸危险的环境中使用。

我们来考察一下，想要用气动执行机构将零件从旋转圆台转送到附近的传送带上的传送装置(参见图1)。

这一装置就是使各执行机构顺序动作的顺序控制装置。

顺序控制装置与基本的计算机的控制思路是一致的，适用于计算机控制的入门。

I 装置的构造

装置是由图 1 所示的四个气动执行机构组装而成的。

执行机构组件

执行机构组件由气动执行机构、换向(电磁)阀与两个位置传感器构成(参见图 2)。

① 手爪伸缩组件

- 汽缸 A:电磁阀 A。

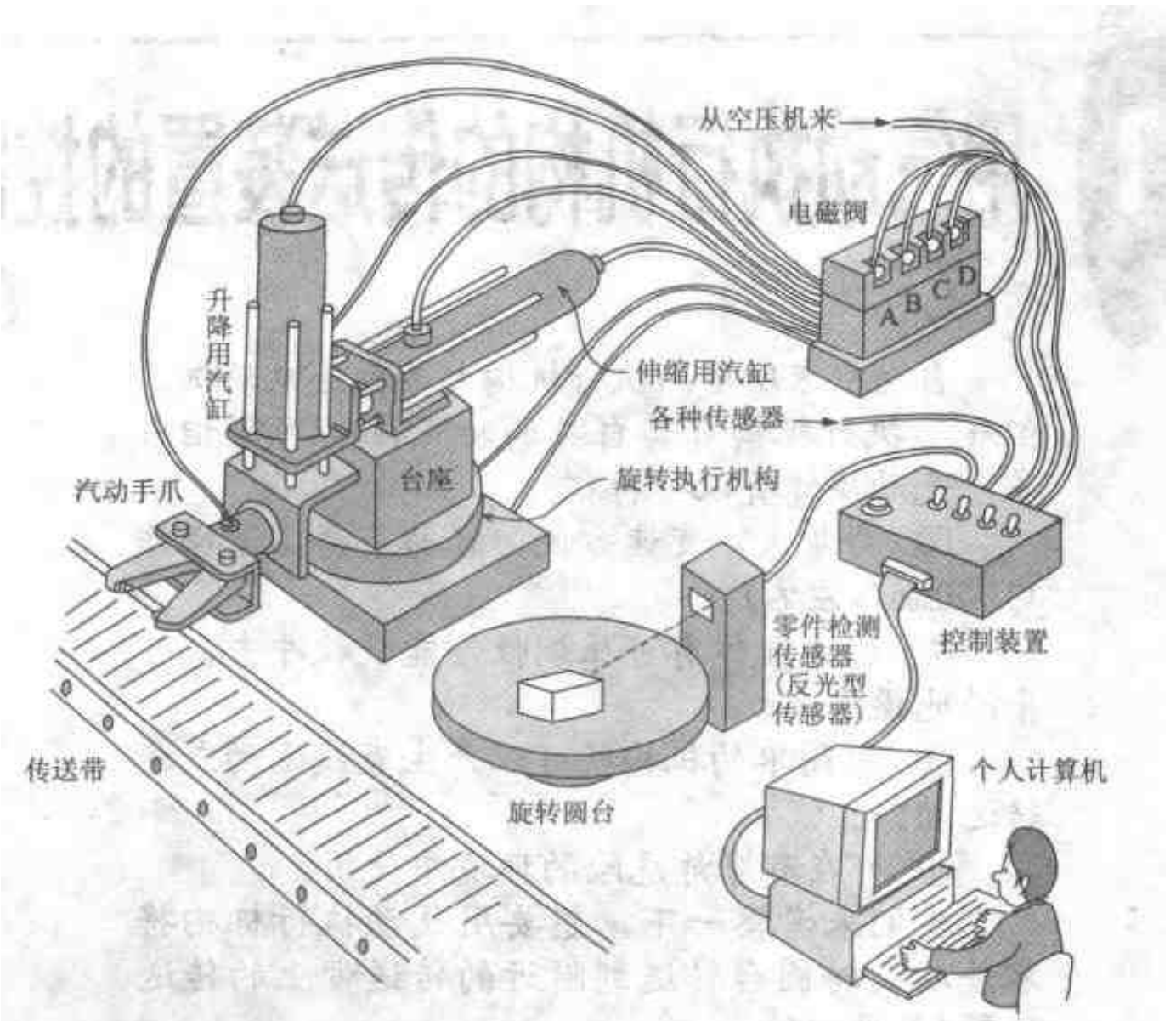


图 1　传送装置简图

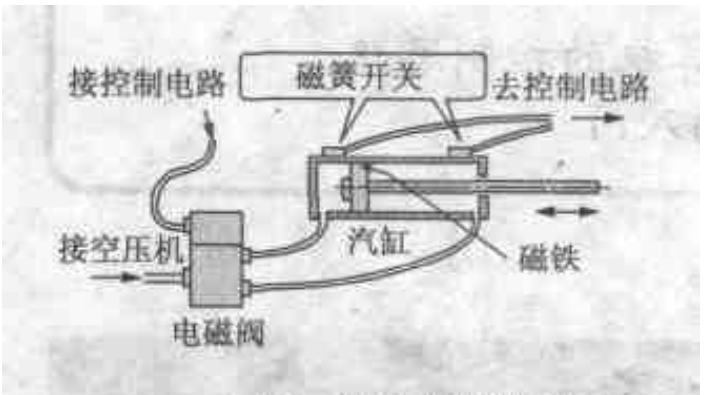


图 2　执行机构组件

• 位置传感器：SA1、SA2(磁簧开关与磁铁)。

② 手爪的升降组件

• 汽缸 B：电磁阀 B。

• 位置传感器：SB1、SB2(磁簧开关与磁铁)。

③ 手爪的转动组件

• 手爪旋转用：电磁阀 C。

• 位置传感器：SC1、SC2(磁簧开关与磁铁)。

④ 手爪

• 手爪执行机构：电磁铁 D。

• 位置传感器：SD1、SD2(磁簧开关与磁铁)。

控制装置

控制装置由执行机构控制电路、开关传感器电路与个人计算机接口电路等构成，控制要求如下：

① 主要动作

· 装置的启动与停止。当零件检测传感器或启动开关当中任意一个发出信号，就用 ON 进入启动过程，过程结束后，即等待启动输入。

· 执行机构的动作。用来自个人计算机的信号，使电磁阀动作，控制汽缸的进气方向，操作执行机构。

通过装在活塞上的磁铁，检测执行机构的端部位置，使汽缸两端的磁簧开关动作。将此信号输入个人计算机。

② 电磁阀驱动电路

来自个人计算机的信号，如图 3 所示，传输到晶体管 Tr_1 与 Tr_2，使电磁阀动作。用信号“1”使电磁阀通电动作，用信号“0”使电磁阀断电返回到起始状态。

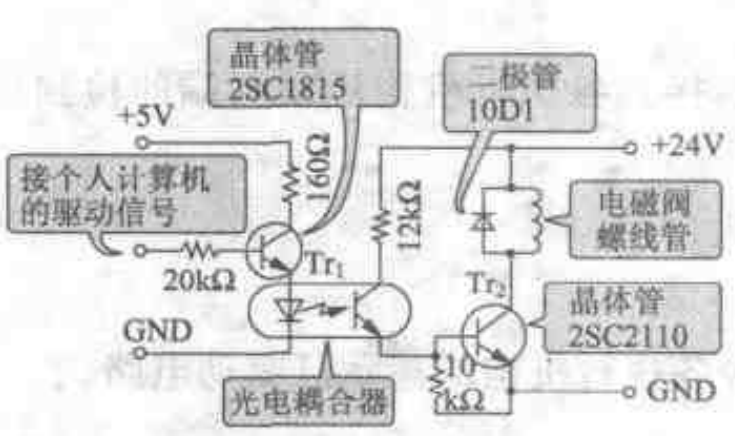


图 3　执行机构驱动电路

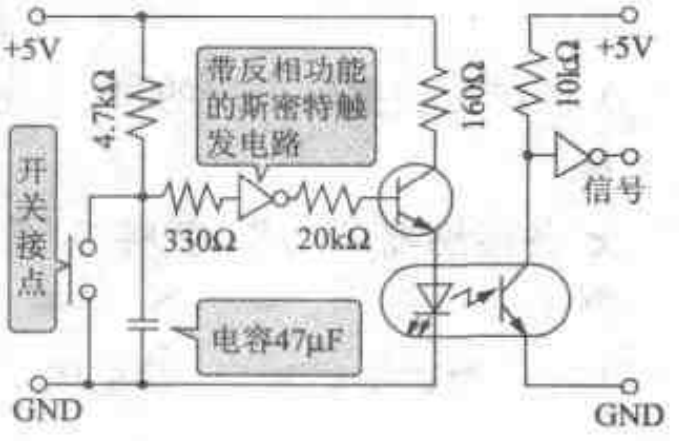


图 4　开关电路

③ 传感器开关电路

用磁簧开关来检测汽缸的位置。复位型接点要用波形整形电路，通过光电耦合器输入计算机的输入输出接口中(参见图 4)。

Ⅱ　与个人计算机的连接

将 8255 输入输出口连接到个人计算机的扩展槽中，用模式 0 设定 A 口与 C 口作为输入用，B 口作为输出，各单元的电路连接如下：

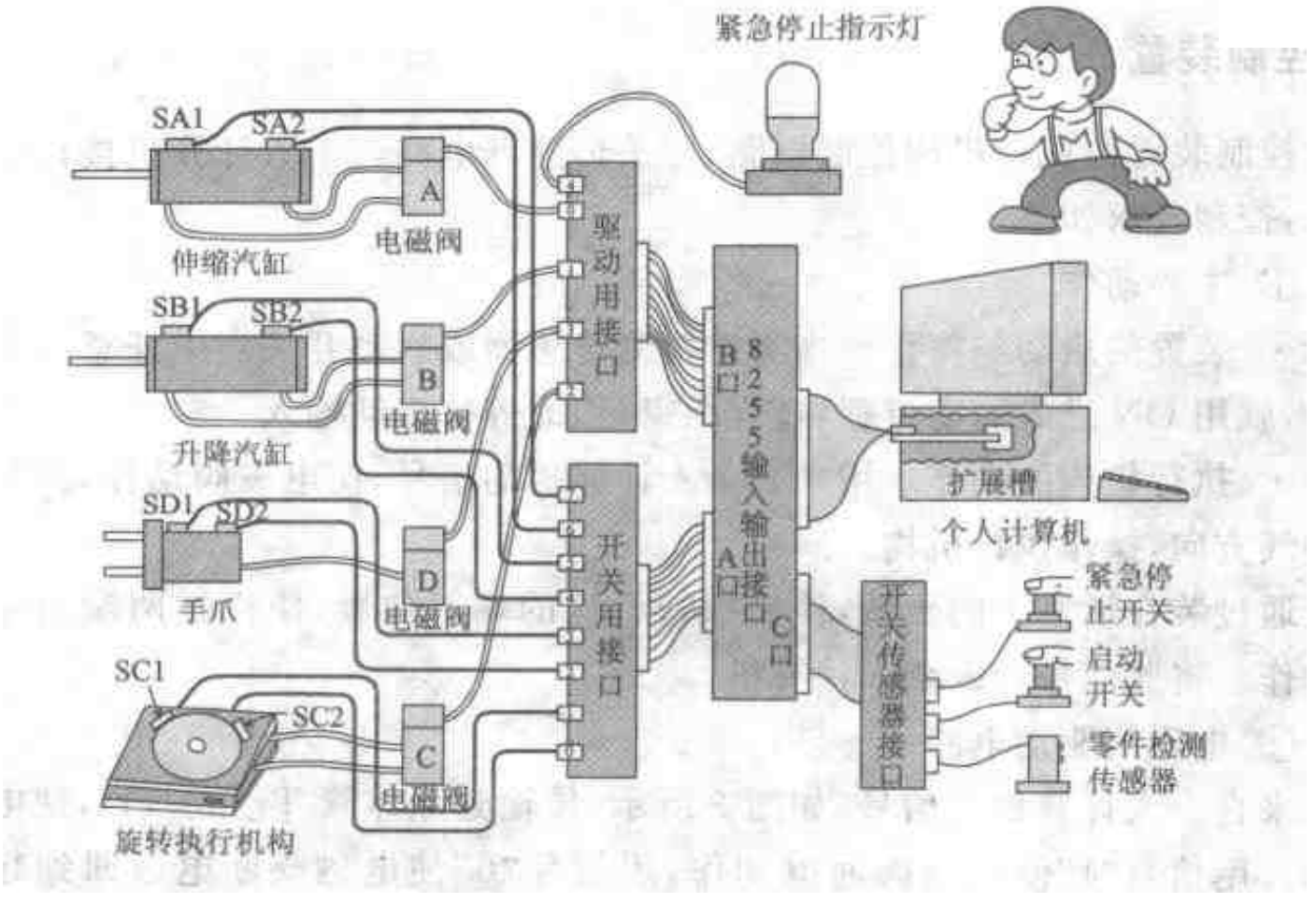


图 5　配置图

① A 口

A 口作为数据输入用的口(参见图 6),接到各执行机构运动两端的检测传感器上。

★ 各传感器若动作,就将“1”信号输入口中。

② B 口

B 口作为数据输出口(参见图 7),连接各执行机构的指示灯驱动电路。

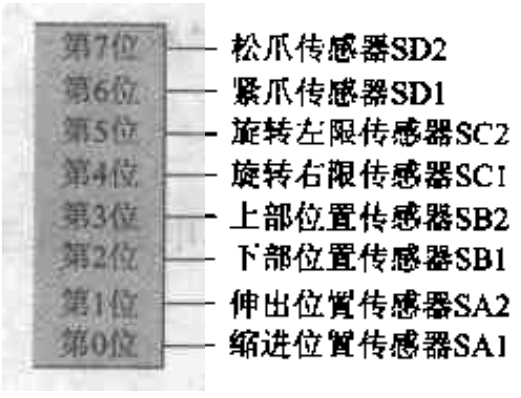


图 6　A 口连接图

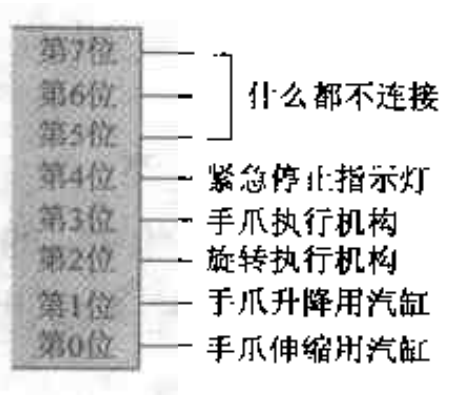


图 7　B 口连接图

- 手爪用信号 1 夹住,用信号 0 松开。
- 旋转执行机构用信号 1 右转,用信号 0 左转。
- 升降汽缸用信号 1 下降,用信号 0 上升。

• 伸缩汽缸用信号 1 伸出，用信号 0 缩回。

③ C 口

C 口用作数据输入口（参见图 8），连接开关或传感器的信号。



图 8　C 口连接图

Ⅲ　动　作

将产品从转台往传送带上转移的作业，要求按以下顺序依次反复控制。

启动——手爪伸出

① 确认零件是否在抓住的位置或启动开关是否已按下。

② 使电磁铁 A 动作，将伸缩汽缸伸出。

手爪下降

① 用位置传感器 SA2 确认伸出状态。

② 使电磁气阀 B 动作，将汽缸 B 下降。

抓住制品

① 用传感器 SB1 确认手爪处于下方。

② 使电磁阀 D 动作，用夹具执行机构将制品抓住。

夹具上升

① 用传感器 SD1 确认 C 抓住制品。

② 切断电磁阀 B 的电源，升降汽缸上升。

手爪缩回

① 用传感器 SB2 确认手爪处于上方。

② 切断电磁阀 A 的电源，伸缩汽缸缩回。

手爪左旋

① 用传感器 SA1 确认手爪处于缩回状态。

② 使电磁阀 C 动作，用旋转执行机构 C 使手爪右旋。

手爪伸出

① 用传感器 SC1 确认手爪处于右旋终端。

② 使电磁阀 A 动作，伸出伸缩用汽缸。

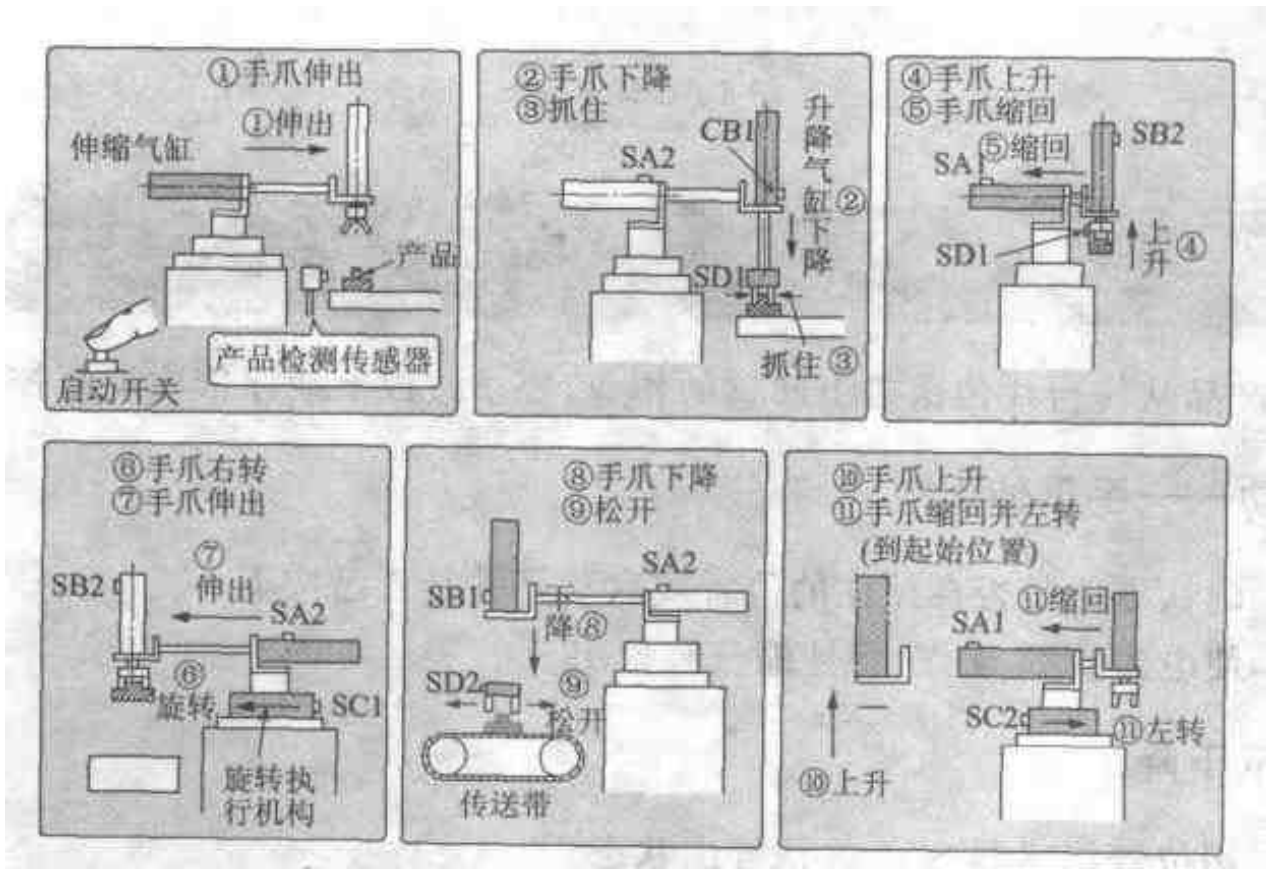


图 9　动作示意图

手爪下降

① 用传感器 SA2 确认手爪处于伸出位置。

② 切断电磁阀 B 的电源，降下升降汽缸。

放下制品

① 用传感器 SB1 确认手爪处于下方。

② 切断电磁阀 D 的电源，松开手爪执行机构，放下制品。

手爪上升

① 用传感器 SD2 确认手爪已松开。

② 切断电磁阀 B 的电源，升降汽缸上升。

缩回手爪并左旋

① 用传感器 SB2 确认手爪处于上方。

② 切断电磁阀 A 与电磁阀 C 的电源，使伸缩汽缸缩回，同时使旋转执行机构右旋，返回起始位置。

★ 随时随地确认紧急停止开关的输入，若有输入时，则将紧急停止信号输出到上一级的控制装置，结束程序。

操作的输入输出信号示于表 1

表 1 各行程的输入输出数据表

操作行程	行程开关启动条件输入数据(输入口) A口 7	6	5	4	3	2	1	0	C口 7	6	5	4	3	2	1	0	行程开关输出数据(输出口) B口 7	6	5	4	3	2	1	0
	抓紧手爪	松开手爪	右旋	左旋	下降	上升	伸出	缩回						紧急停止	启动开关	零件检测				紧急停止	松0紧1	右旋1左旋0	升0降1	伸1缩0
①启动、手爪伸出	第1个行程的启动条件，是C口的输入数据								零件检测传感器				0	0	0	1	0	0	0	0	0	0	0	1
													(1)16				(1)16							
②手爪下降	0	1	0	1	0	1	1	0	操作启动开关				0	0	1	0	0	0	0	0	0	0	1	1
	(56)16												(2)16				(3)16							
③爪住工件	0	1	0	1	1	0	1	0	操作紧急停止开关				0	1	0	0	0	0	0	0	1	0	1	1
	(5a)16												(4)16				(b)16							
④手爪上升	1	0	0	1	1	0	1	0									0	0	0	0	1	0	0	1
	(9a)16																(9)16							
⑤手爪缩回	1	0	0	1	0	1	1	0									0	0	0	0	1	0	0	0
	(96)16																(8)16							
⑥手爪右旋	1	0	0	1	0	1	0	1									0	0	0	0	1	1	0	0
	(95)16																(c)16							
⑦手爪伸出	1	0	1	0	0	1	0	1									0	0	0	0	1	1	0	1
	(a5)16																(d)16							
⑧手爪下降	1	0	1	0	0	1	1	0									0	0	0	0	1	1	1	1
	(a5)16																(f)16							
⑨松开工件	1	0	1	0	1	0	1	0									0	0	0	0	0	1	1	1
	(aa)16																(7)16							
⑩手爪上升	0	1	1	0	1	0	1	0									0	0	0	0	0	1	0	1
	(6a)16																(5)16							
⑪手爪缩回并左旋90度	0	1	1	0	0	1	1	0									0	0	0	0	0	0	0	0
	(66)16																(0)16							



Ⅳ 将以上的动作编制为程序

① 程序 1

由于“从确认输入信号→执行装置的动作”这一连串操作要反复 11 次，可

作为一连串的动作来设计程序。但第一行程有两种不同的启动输入。

① 将输入口(地址＝wwww)的数据输入变量 indt 中。

② 要将变量 indt 的输入值与必要的输入数据(xxxx)比较是否相同。

③ 不同则反复执行②、③。

④ 若相同则将使执行机构动作的数据(yyyy)输出到 B 口(地址＝$(d2)_{16}$)。对此用 do-while 语句写成如下程序：

```
do{
    indt=inp(wwww);
}while(indt!=xxxx);
outp(0xd2,yyyy);
```

② 设置变量

由于口地址与数据是根据操作不同而不同的，设置变量是以反复 10 次动作作为结束一件工作，以此作为无条件循环的程序为：

```
for (;;)
 {
   for (i=0;i<=9;i++);
    {
      do{
      indt=inp(0xd0);
      }while(indt!=stdt [i]);
      outp(0xd2,outd [i]);
    }
}
```

在此程序之前，必须编写设置变量的声明并将数据代入变量的程序。

```
设置变量声明与初始化的例子
    unsigned int
outd[10]={0x22,0x33,0x44,0x55,0x66,0x88,0x99,0xaa,
0xbb};
    [ ]表示设置变量的数,从outd[0]起依次代入{ }
内的数据。
```

③ 紧急停止信号的处理

因需要随时对紧急停止信号进行检查，所以一般用中断功能来处理，这里只考虑用程序来处理。输入了紧急停止信号时的处理，点亮紧急停止指示灯，并使程序停止执行。

处理程序如下：

```
1  for (;;)
2  {
3    hj=inp(0xd4);
4    hj=hj&0x4;
5    if (hj= =0x4)
6      {
7            outp (0xd4,0x10);
8            break;
9      }
10 }
```

④ 程序

将以上的程序合成一个程序，如下页所示，在第 7 章第 5 节中，汽缸控制所用的是 switch-case 语句，在这里则用设置变量与 for 语句。

⑤ 程序说明

4～5 中进行两个变量的设置与初始值的设定。

6～31 的 for(;;)为对全行程(11 行)进行无条件循环。当用 20～26 检测到紧急停止信号时，则跳出循环，结束程序。

17～30 行 for(i=0;i<=9;i++)是将 19～29 行执行 10 次，根据设置变量而按不同的行程执行。

19～28 行 do-while 语句是对紧急停止信号的监视处理和行程启动输入的检查。若有启动行程的数据输入，则用 29 行将该行程操作数据输出。

21 行按位逻辑与运算，用于要提取特定位时。

由于紧急停止的检测必须是独立的，与其他输入信号无关，所以要提取第 2 位的信号，以 0x4(0000 0100)为输入数据进行按位逻辑与运算。

25 行 goto 语句，跳转到标记为 owari(结束)处，跳出三重循环圈，结束程序的运行。

前页的程序是用 break 语句跳出循环的，因有三重循环，所以用 goto 语句跳出。

★ 标记用(冒号)":"设置，置于执行语句的开头或单独一行中。

⑥ 汽缸传送装置控制程序

```
1  main()
2  {
3      unsigned int i,hj,sns;
4      unsigned int stdt[10]={0x56,0x5a,0x9a,0x96,0x95,0xa5,
           0xa6,0xaa,0x6a,0x66};
5      unsigned int outd[10]={0x3,0xb,0x9,0x8,0xc,0xd,0xf,0x7,
           0x5,0x0};
6    for(;;)
7    {
8        for(;;)
9        {
10           sns=inp(0xd4);
11           if(sns==0x1||sns==0x2)
12           {
13               break;
14           }
15       }
16       outp(0xd2,0x1);输出第一行程的动作数据
17       for(i=0;i<9;i++)
18       {
19           do{
20               hj=inp(0xd4);
21               hj=hj&0x2;
22               if(hj==0x2)
23               {
24                   outp(0xd4,0x10);
25                   goto owari;
26               }
27               sus=inp(0xd0);
28           }while(sns!=stdt[i]);
29           outp(0xd2,outd[i]);
30       }
31   }
32 owari:;
33 }
```

各行程启动信号的输入数据

各行程输出数据的地址

配置变量的定义与初始化

在零件传感器或启动开关操作之前，对输入进行无条件地循环判断

无条件循环的for语句，有紧急停止输入信号的场合，跳出循环圈

依次代入第10行的设置变量，执行10次行程

监视紧急停止信号的输入，有输入的情况下，输出紧急停止指示灯的亮灯数据并结束程序

在位置传感器信号的输入数据等于行程的启动数据之前，反复地进行判断



2 简单自动门的控制

说句“芝麻开门”，可以使山洞的石门打开，以前是靠魔法的力量，现在则要用声音传感器系统与判断语言的电动执行机构来考察一下，能否使石门打开。自动门大多用在进出人流大的场所。自动门应用的是什么样的机构？怎样进行控制？这里对有关自动门的课题作一下探讨，并假设一个试验用的自动门，编制对它进行控制的程序。

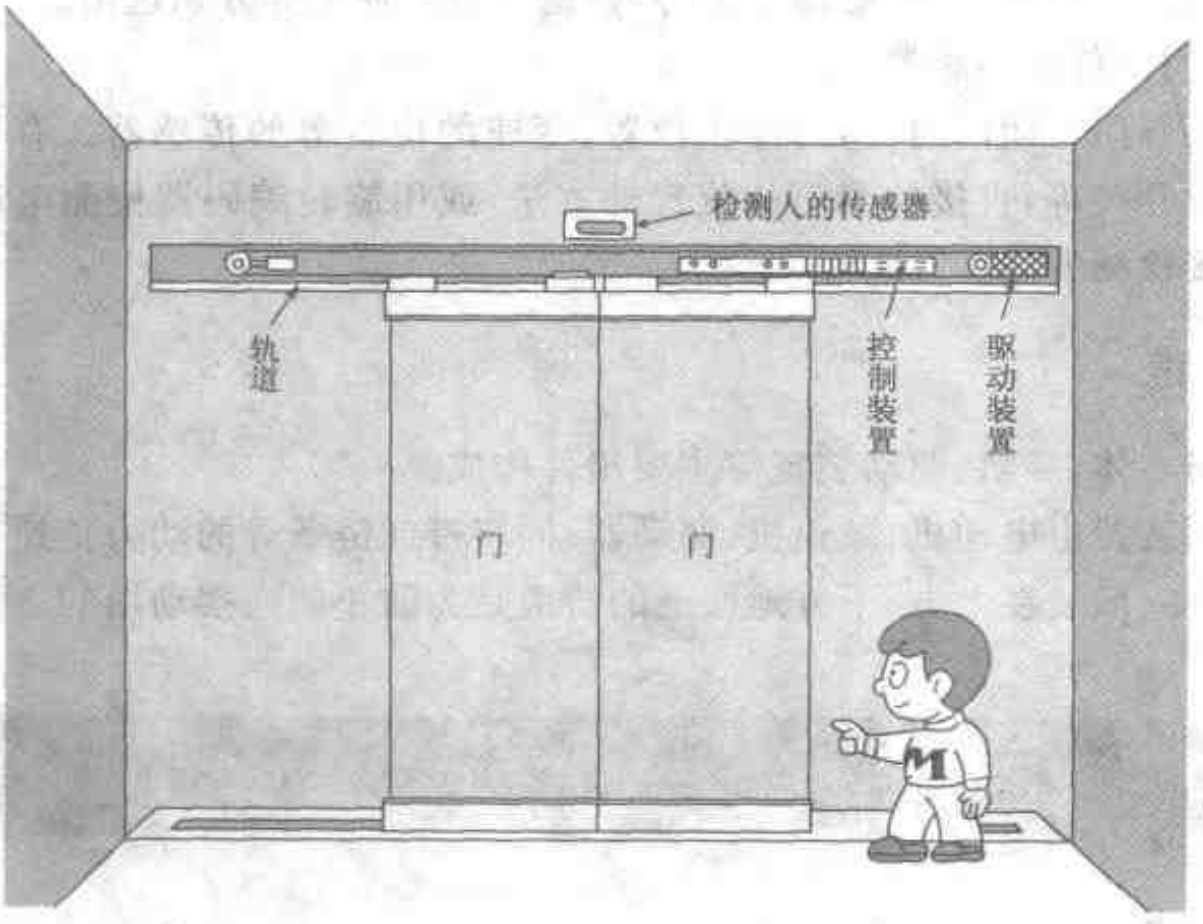


图 1　自动门

Ⅰ　自动门的构造

动力源与执行机构

一般的自动门是由某种动力源与执行机构组成的(参见图 1)。

① 动力源

尽管动力源有电气、液压、气动装置等,但一般建筑物的自动门都用电气动力源。

② 执行机构

尽管执行机构根据动力源有电动机、油缸、汽缸等,但一般采用电动机。

传感器

用来检测分析门前有没有人,门正在开还是正在关等状态。

① 检测人的传感器

检测人的传感器有超声波传感器、热电传感器(感热型),光传感器(透射型、反射型)。罩面开关、触摸开关等,根据使用的环境等分别选用。

② 门的位置传感器

确认开门与闭门时,门的停止位置、变速的位置等的传感器。有用限位开关或接近开关等,直接检测门的位置的方法,或用旋转编码器检测电动机的转数的间接检测方法。

构　造

由门本体、导轨、驱动装置等主要组件构成。

驱动装置由电动机、减速机、制动器、同步带或链条等的动力传递部件等构成。主轨一般装在上方,下方地面上的轨道是为防止侧向摆动用的。

Ⅱ　控制假想的自动门

构　造

假想的自动门控制如图 2 所示

① 控制动作

检测到人时就开门,确认没有人,则过一定时间后闭门。开门动作从低速

到高速，通过二级加速度到达正常速度，在停止之前，通过二级减速度，并用制动器制动。

② 采用单方向开门方式

在上部装有无侧向松动的轨道。

③ 驱动方式

使用带减速机与制动器的直流电动机(额定值为 24V、50W)。用同步带与同步带驱动轮向门传递动力。

④ 人的检测

用设置在门内外侧的超声波传感器，检查有没有人。

⑤ 停止位置与开始位置

在闭门端与开门端的低速运行开始位置与停止位置，均设置位置检测用的限位开关。

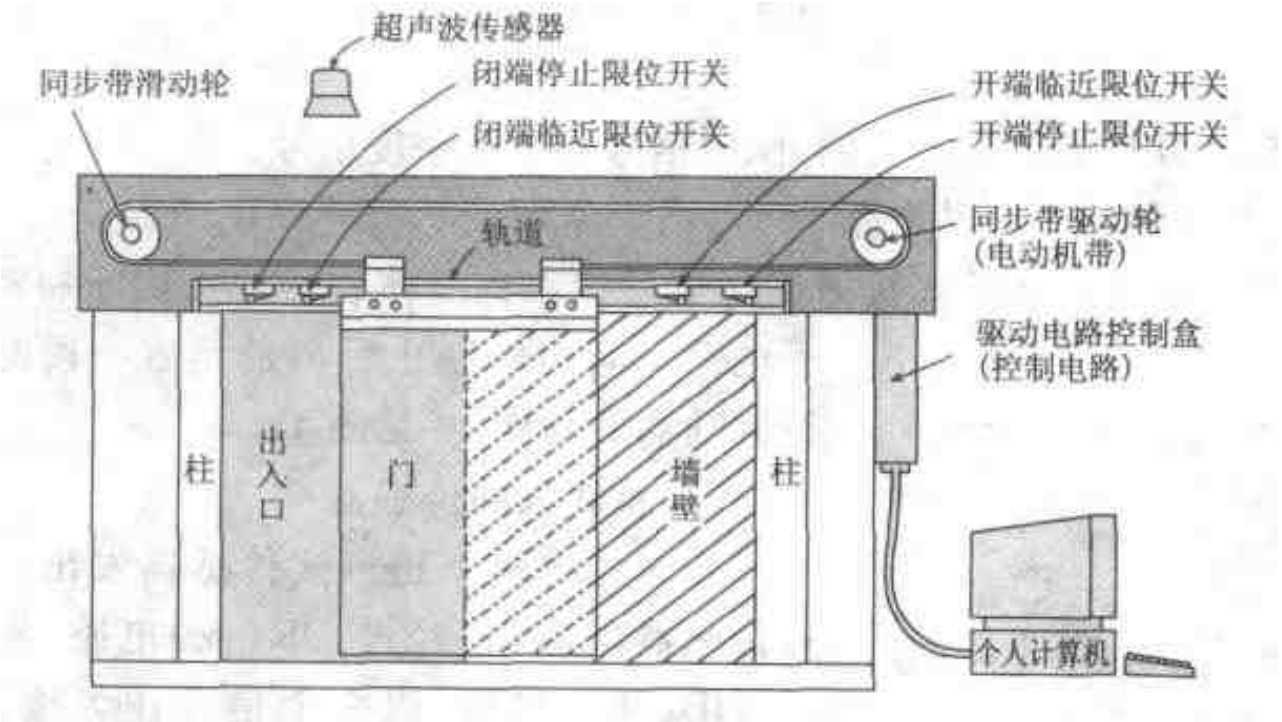


图 2 假想的自动门

控制电路与个人计算机的连接

① 个人计算机端的接口

使用 8255 并行数据输入输出接口的模式 0，设定 A 口为输入口，B 口为输出口(参见后面"口连接"部分)。

② 电动机驱动电路

开、闭门的电动机如用图 3 所示，正(开)、反(闭)转时使用 4 个晶体管 Tr_1、Tr_2、Tr_3 及 Tr_4 作为驱动控制，用晶体管 Tr_1、Tr_2 的基极电压与电阻作为三级

变速控制，来分级改变施加到电动机上的电压。

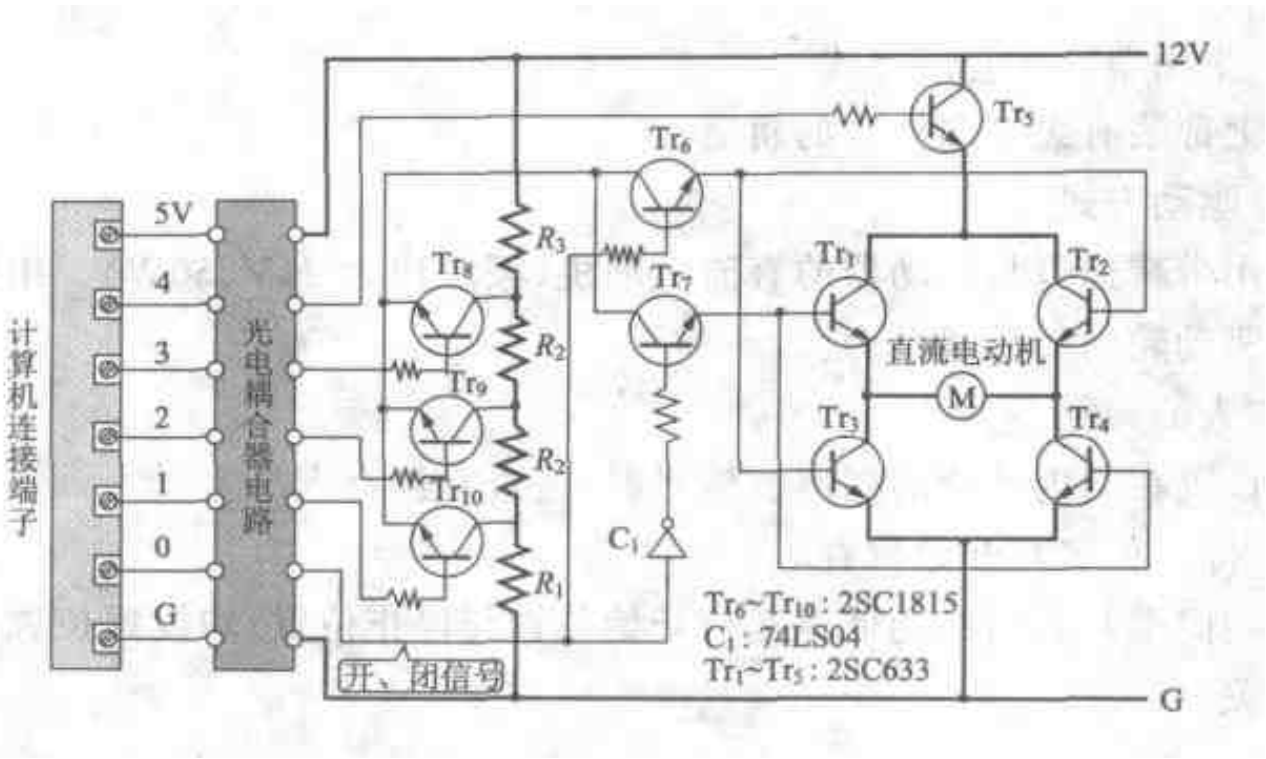


图 3 电动机驱动电路

来自计算机的信号是门的开、闭信号和三个速度信号。

③ 制动器动作驱动电路

制动器操作是将计算机的制动信号输入到晶体管，接通、切断制动器的驱动电路。电动机停止后稍稍延迟一段时间再进行制动器的动作(参见图 4)。

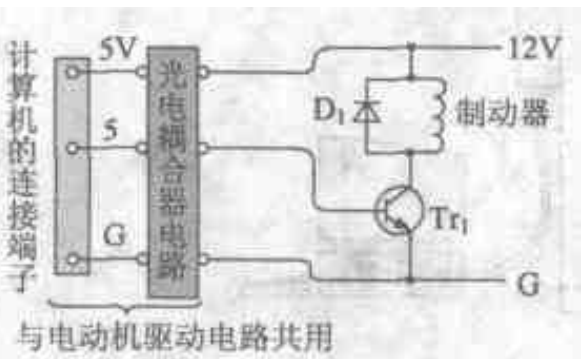


图 4 制动器动作驱动电路

④ 检测人的传感器

门内外的两个超声波传感器发出的信号经过传感器电路，再经过 OR(或)电路(实际使用 NAND(与非))后，以一个信号作为输入(参见图 5)。

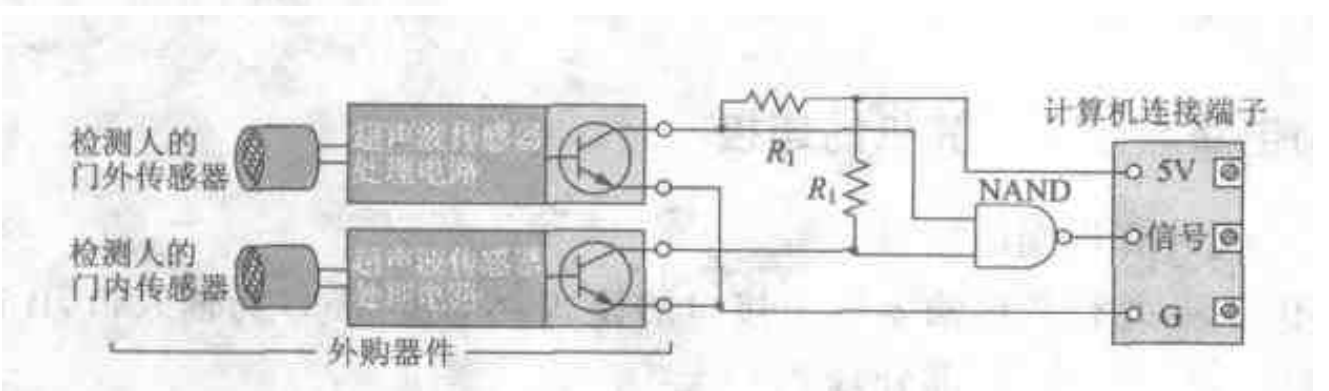


图 5 检测人的传感器

⑤ 开关输入

限位开关或按钮开关的信号，通过信号波形整形电路输入。在开门行程和闭门行程中，有临近传感器与停止传感器。

· 临近传感器

临近传感器是检测临近停止位置时要变换门的运行速度用的限位开关。

· 停止传感器

停止传感器是检测使门停止位置的限位开关。

· 紧急开关

紧急开关是在检测人的传感器失灵等场合，对付异常事态用的开关。

I/O口连接

如图 6 所示，将输入装置接到输入输出口。



图 6　I/O口连接图

基本动作

① 从检测人的传感器起作用开始，就对处于范围内的人进行监视。

② 当检测到人的信号时，就开门。

门以低速开 0.5 秒后进入中速，并从临近闭端起，开始常速运行。

③ 若开端的临近检测传感器检测到门，则使开门经过 0.5 秒后，由中速运行进入低速。

④ 当开端的停止传感器检测到门，门就停止运行。

⑤ 门开一段时间，当检测人的传感器确认无人后，闭门。

⑥ 闭门行程也从闭门开始到结束附近，改变运行速度，平滑地运行。

⑦ 在门运行过程中，用检测传感器监视人，若检测到人，则返回开门行程。

⑧ 门的停止运行是切断电动机的电流，约 0.5 秒后，使制动器起作用。开门时，当制动器松开约 0.5 秒后，使电动机运行。

★ 检测人　为了安全，在闭门运行过程中，必须始终用检测人的传感器检测是否有人，检测有人的情况下，立即停止闭门，并运行开门的行程。

Ⅲ　程　序

编制一个汇总这些操作的程序。

定　义

为了编制程序方便，用 # define 语句设置口地址与常用的带名称的输入输出数据，名称为了易于识别，一律用大写英文字母。

■　口地址

```
# define APORT 0xd0…A 口地址
# define BPORT 0xd2…B 口地址
# define CPORT 0xd4…C 口地址
# define CWREG 0xd6…CW 寄存器地址
```

■　输入数据

```
# define HITOKEN 0x10…检测人的数据
# define OPNTEMAE 0x1…开端临近传感器
# define OPNTEISI 0x2……开端停止传感器
# define CLSTEMAE 0x4…闭端临近传感器
# define CLSTEISI 0x8……闭端停止传感器
```

■　输出数据

```
# define OPN 0x19……开门运行
# define OPL2 0x15……  开门中速
# define OPL1 0x13……  开门低速
# define OPNSTP 0x1…  门全开停止
# define CLS 0x18…… 闭门运行
# define CLSL2 0x14…… 闭门中速
# define CLSL1 0x12…… 闭门低速
# define CLSSTP 0x0……门全闭停止
# define BRAKE 0x20……制动
```

函　数

对于反复输入、判断处理的动作，仅数据有变化时，可将共同部分的程序编

人函数中。因根据处理有判断一次与两次的情况，可编制两个判断处理函数（参见图 7）。

人的检测在闭门运行过程中或在闭门状态下都是必要的，并且此检测与其他的检测不发生关系，所以要将输入数据与人的检测数据进行按位逻辑与（AND）运算，仅提取人的检测数据。

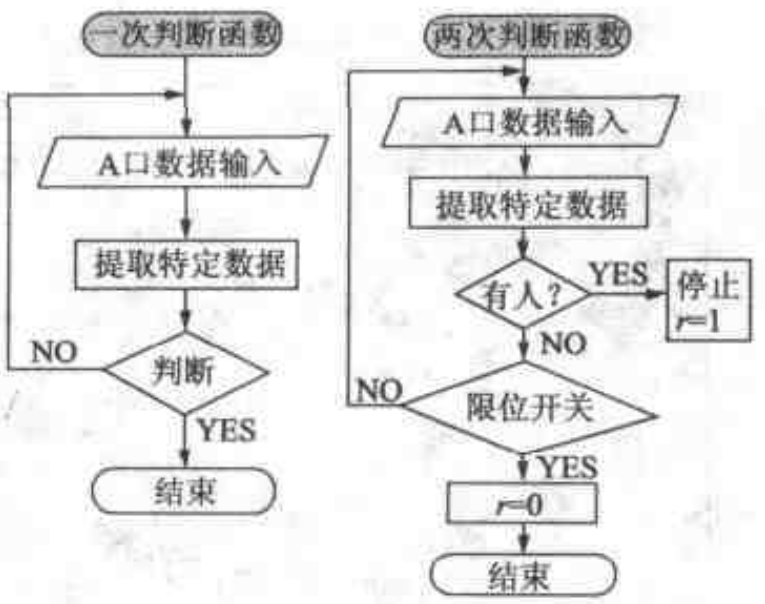


图 7　函数流程图

函数程序

函数-1　检测人的传感器，对是否检测到人进行判断等，作为一次判断函数程序，用 do-while 语句编写。

函数-1程序<一次判断>

```
1  handan1(unsigned int nyur) ———— nyur是形参，用于通过调用函数程序来输入数据。
2  {
3        unsigned int date;
4        do{ ————————————————
5              date=inp(APORT)&nyur;
6        }while(date !=nyur); ————
7  }
```

do-while语句，用于当变量date的内容不等于形参nyur的内容时，进行反复地循环处理。(处理：将A口的输入数据与形参nyur进行运算，并代入变量date中。)当do-while语句不成立(变量date的内容与形参nyur的内容相等时)的场合，将此函数程序结束。

函数-2　将检测人的传感器与门的位置传感器，作两次判断与处理的函数程序，编制成用无条件循环的 for(;;)语句与 if(　)语句编写的函数程序。

函数-2的程序<两次判断>

```
1  handan2(unsigned int nyu1);
2  {
3        unsigned int x,h,r;
4        for(;;)
5        {
6              x=inp(APORT);
7              h=x&HITOKEN;
8              if(h= =HITOKEN)
9              {
10                   outp(BPORT,OPNSTP);
11                   r=1;retune(r)
12                   break;
13             }
14             else if (x= =nyu1)
15             {
16                   r=0;retune(r)
17                   break;
18             }
19       }
20 }
```

无条件循环。

从12:与17:的break语句中跳出。

将A口的输入数据代入变量x中。

<检测人>
当变量h的内容与识别值HITOKEN(检测到人的数据)相等时，将识别值OPNSTP(停止)输出到B口，将1返代入变量r中，于是用break语句跳出for(;;)语句的无条件循环，函数程序结束。

<传感器的检测>
当变量x的内容与形参nyu1相等时，将0返代入变量r中，于是用break语句跳出for(;;)语句的无条件循环，函数程序结束。

使用本节的定义了函数，编制的 main 函数程序：

```
1   main()
2   {
3           unsigned int in,in1,in2,in3,ot,ot1,ot2,r;
4           outp(cwreg,0x90);
5           for(;;)
6           {
7                   handan1(HITOKEN);
8   hiraki:         outp(BPORT,OPNL1);
9                   for(t= =1;t<=50;t++)
10                  {
11                      handan1(CLSTEMAE);
12                  }
13                  outp(BPORT,OPNL2);
14                  handan1(CLSTEMAE);
15                  outp(BPORT,OPN);
16                  handan1(OPNTEMAF);
17                  outp(BPORT,OPNL2);
18                  for(t= =1;t<=50;t++)
19                  {
20                      handan1(OPNTEISI);
21                  }
22                  outp(BPORT,OPNL1);
23                  handan1(OPNTEISI);
24                  outp(BPORT,OPNSTP);
25                  for(t= =1;t<=10;t++);
26                  outp(BPORT,BRAKE);
27                  for(t= =1;t<=20000;t++);
28                  do{
29                      d=inp(APORT)&HITOKEN;
30                  }while(d= =HITOKEN)
31                  outp(BPORT,0x0);
32                  for(t= =1;t<=10;t++);
33                  outp(BPORT,CLSL1);
34                  for(t= =1;t<=50;t++)
35                  {
36                      handan2(HITOKEN,OPNSTP,OPNTEMAE);
37                  }
38                  if(r= =1)
39                  {
40                      goto hiraki;
41                  }
42                  outp(BPORT,CLSL2);
43                  for(t= =1;t<=50;t++)
44                  {
45                      handan2(HITOKEN,OPNSTP,OPNTEMAE);
46                  }
47                  if(r= =1)
48                  {
49                      goto hiraki;
50          }       outp(BPORT,CLS);
51                      handan2(HITOKEN,OPNSTP,CLSTEMAE);
52                  if(r= =1)
53                  {
54                      goto hiraki;
55                  }
56                  outp(BPORT,CLSL2);
57                  for(t= =1;t<=50;t++)
58                  {
59  handan2(HITOKEN,OPNSTP,CLSTEISI);
60                  }
61                  if(r= =1)
62                  {
63                      goto hiraki;
64                  }
65                  outp(BPORT,CLSL1);
66  handan2(HITOKEN,OPNSTP,CLSTEISI);
67                  if(r= =1)
68                  {
69                      goto hiraki;
70                  }
71                  outp(BPORT,CLSSTP);
72                  for(t= =1;t<=10;t++);
73                  outp(BPORT,BRAKE);
74          }
75  }
```

将实参代入检测值，调用函数handan1

低速运行开门

在校验闭端临近传感器起作用的同时，保持低速1运行一段时间

中低速运行开门

校验闭端临近传感器的作用

正常速度运行开门

校验闭端临近传感器的作用

中速运行开门

在校验开端停止传感器起作用的同时，保持中速运行一段时间

低速运行开门

校验开端停止传感器的作用

使门停止运行

定时间后，使制动器动作

定时间内，保持开门状态

确认没有人

松开制动器

定时间后，门开始运行，低速运行闭门

在校验检测人与开端临近的传感器起作用的同时，低速运行一段时间

在r中返代入1的场合，执行标记为hiraki的语句。(检测有人的情况)

中速运行闭门

在进行人的检测与开端临近传感器起作用检测的同时，中速运行一段时间

将1返代入r的情况下，执行标记为hiraki的语句

正常速度运行闭门

将1返代入r的情况下，执行标记为hiraki的语句

中速运行闭门

在进行人的检测与闭端停止传感器起作用检验的同时，中速运行一段时间

将1返代入r的情况下，执行标记为hiraki的语句

低速运行闭门

检验检测人的传感器与闭端停止传感器的作用

将1返代入r的情况下，执行标记为hiraki的语句

停止闭门

定时间后，使制动器起作用



3 用步进电动机的传送装置的控制

步进电动机有着低功耗、可频繁启动、停止的特点，已作为驱动电动机广泛用于打印机、硬盘存储装置等计算机的外围设备上。

现在设置一台用步进电动机的传送装置，设计一下它的控制程序。

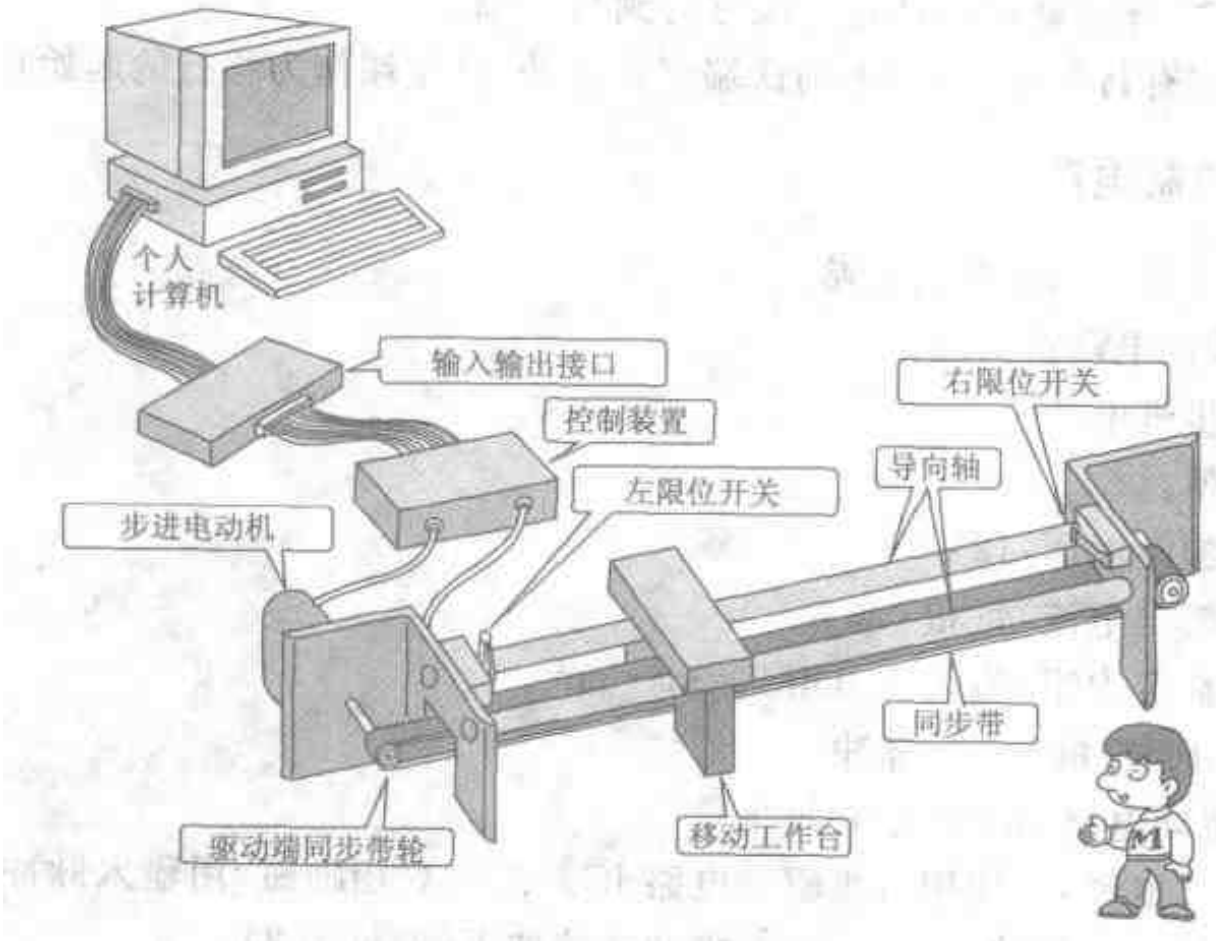


图 1　传送装置构造图

I 步进电动机传送装置

如图 1 所示，设计一个直线运动的移动工作台。

构　造

通过同步带轮与同步带，将步进电动机的旋转运动转化为直线运动，使移动工作台左右移行，移动工作台用两根笔直的导向轴进行导向。

使用同步带轮与同步带，是为了使电动机的旋转精确无滑动地传送到移动工作台。

使用每个脉冲 1.8°的步进电动机，同步带的齿距为 2.0mm，同步带轮的齿数为 20 的话，则每一转可使移动工作台移动

$$2.0\times20=40.0$$

即 40mm。步进电动机每 200 个脉冲旋转一周，则每一个脉冲的移动量为

$$40.0\div200=0.2$$

即 0.2mm。也就是说，此传送装置的控制精度是 0.2mm。

若最大移动量为 300mm。

$$300\div0.2=1500$$

即用 1500 个脉冲就可以从这一端移行到另一端。

两端装有行程开关，用于确认端部。另外，以左端作为移行的起始点。

驱动控制电路

(1) 步进电动机控制电路

① 步进电动机的规格

双相步进电动机

额定值：

额定电压：12V

额定电流：每相 0.3A

额定力矩：最大静力矩 2.2kg·cm

步 进 角：1.8°/脉冲

② 驱动电路 IC(东芝 TD62803T)

为简化电路，可使用市售驱动电路 IC。使用双相励磁，用输入脉冲信号与转向信号的方法驱动。主要 IC 引脚的连接如下(参见图 2)。

引脚 1：转向信号输入(高(1)右转、低(0)左转)。

引脚 2：高(1)
引脚 3：低(0) } 设定双相励磁驱动
引脚 4：低(0)

引脚 6：电动机的黑色线
引脚 7：电动机的红色线
引脚 9：电动机的绿色线
引脚 10：电动机的蓝色线
} 驱动电流流向地。

引脚 14：脉冲信号输入。

（流入的最大驱动电流，每一引脚为 0.4mA。）

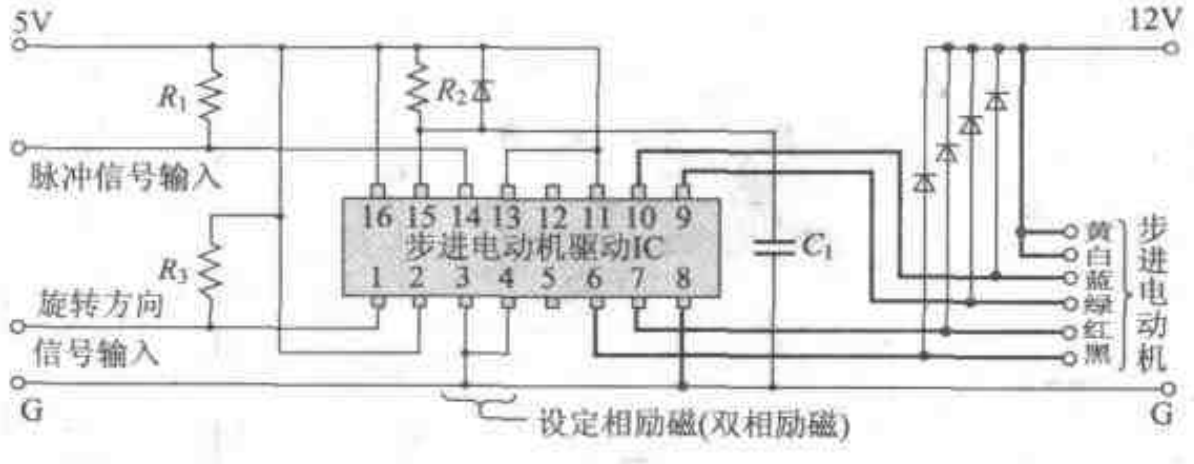


图 2　步进电动机驱动电路 IC

③ 脉冲信号

脉冲信号可通过脉冲发生电路与通过计算机两种方法来产生。在以固定频率驱动步进电动机的场合，使用脉冲发生电路，而在使用各种不同频率来驱动步进电动机的场合，则用计算机来产生脉冲。

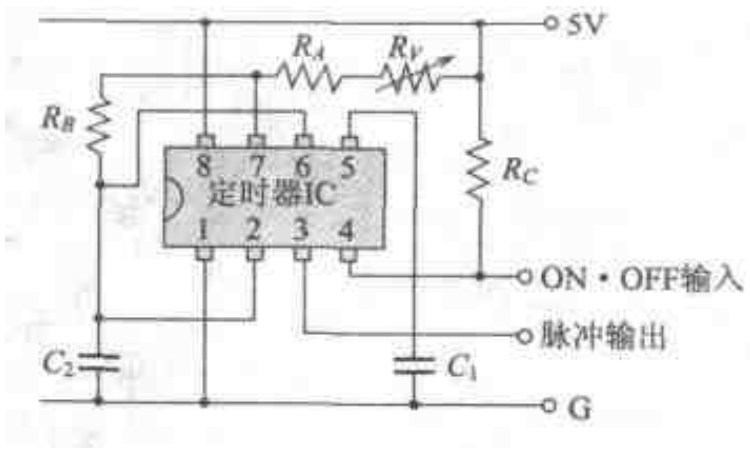


图 3　脉冲发生电路

④ 脉冲发生电路

用定时电路 IC(555)来制作产生脉冲的电路(多谐振荡器)，通过半可调电阻 R_v 来调节脉冲频率(参见图 3)。

引脚 4 设置为高(1)时产生脉冲，而低(0)时停止产生脉冲。

脉冲信号从引脚 3 输入。

(2) 控制电路

① 计算机控制与手动操作的选择

转向信号用开关 SW_1，脉冲信号用开关 SW_2，是否用来自计算机的信号，通过手动开关来选择(参见图 4)。

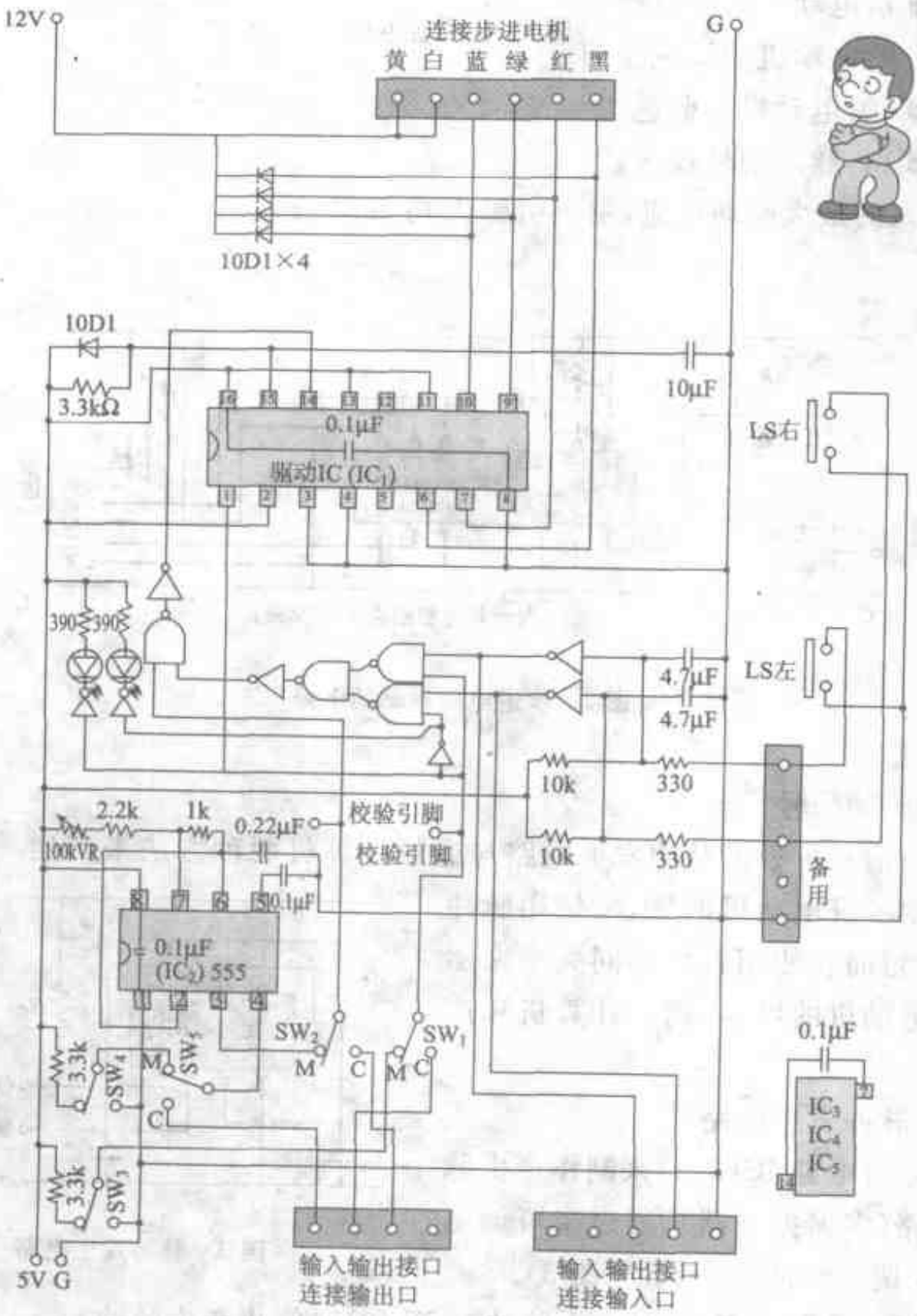


图 4　步进电动机控制电路

② 手动操作

· 将开关 SW_1、SW_2 设定在手动操作位置(参见图 5)。

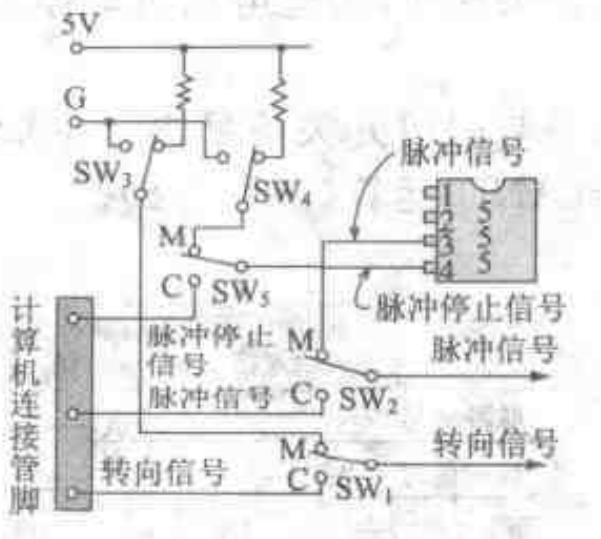


图 5　手动操作开关的设定

- 用开关 SW_3 设定转向。
- 用开关 SW_4 使脉冲停止。

③ 计算机控制

可使用以下两种方法：

- 方法 1

将开关 SW_1、SW_2 设置为计算机控制位置，使用来自计算机的脉冲信号与转向信号(参见图 6)。

- 方法 2

将开关 SW_1、SW_5 设定在计算机控制位置，将开关 SW_2 设定到手动操作位置。于是，转向信号与脉冲停止信号用来自计算机的信号，而脉冲信号则用来自脉冲发生电路的信号(参见图 7)。

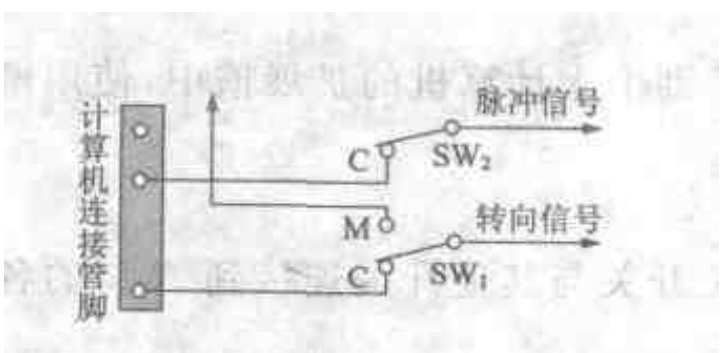


图 6　计算机控制的开关设置 1

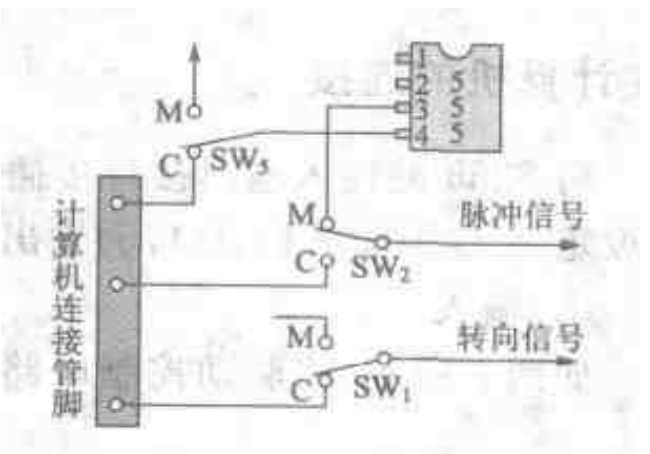
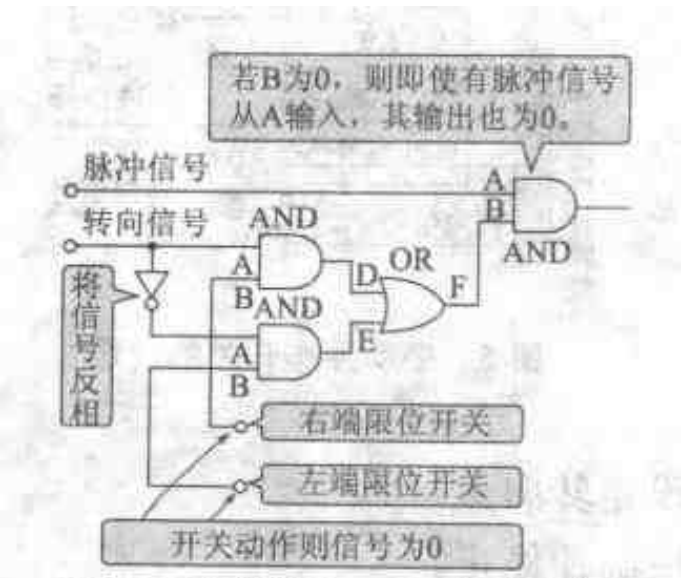


图 7　计算机控制的开关设置 2

④ 两端的界限处理

若移行两端的限位开关动作，步进电动机就停止移行，但离开限位开关的方向仍可继续移行。

其原理是用转向信号与限位开关信号进行逻辑与(AND)运算，其结果再与脉冲信号进行逻辑与(AND)运算(参见图 8)。



AND(与)电路真值表

输入		输出
A	B	
0	0	0
1	0	0
0	1	0
1	1	1

OR(或)电路真值表

输入		输出
D	E	F
0	0	0
1	0	1
0	1	1
1	1	1

图 8　移行两端界限处理原理

⑤ 开关电路

两个限位开关与两个按钮开关(启动与返回起始点)的信号，经过波形整形电路输入到计算机中。

计算机的连接

将 8255 的输入输出接口接插板连接到个人计算机的扩展槽中，使用模式 0，设定 A 口为输入口，B 口为输出口。

① 输入

如图 9 所示，将驱动控制电路的限位开关与其他开关连接到 A 口的各管脚上。

② 输出

如图 10 所示，将驱动电路的脉冲信号、移行(旋转)方向信号，产生脉冲的启动信号接到 B 口的各管脚上。

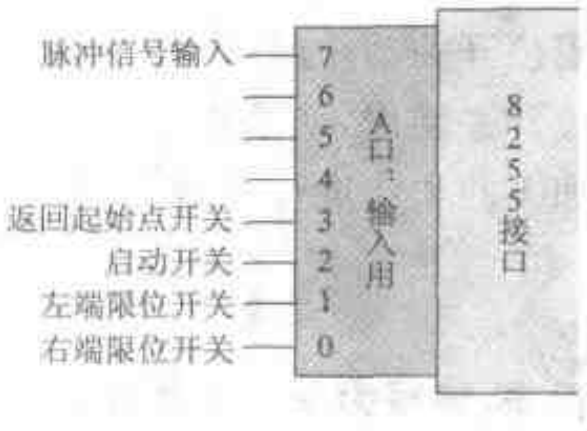


图 9 输入口的连接

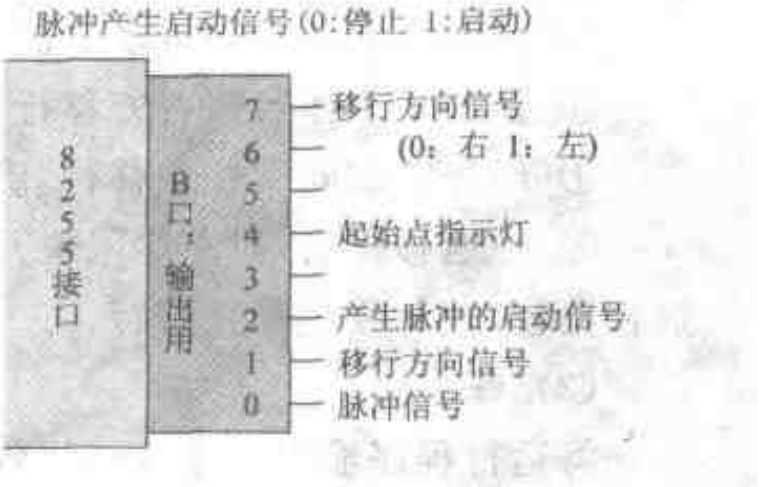


图 10 输出口的连接

定 义

将口地址号置换为口名称。

```
#define APORT 0xd0    A 口地址号
#define BPORT 0xd2    B 口地址号
#define CPORT 0xd4    C 口地址号
#define CWREG 0xd6    CW 寄存器地址号
#define RLS 0x1    右端限位开关数据
#define LLS 0x2    左端限位开关数据
#define RGT 0x0    移行方向向右
#define LFT 0x2    移行方向向左
#define PLSH 0x1    脉冲高电平信号
#define PLSL 0x0    脉冲低电平信号
#define MAXL 300    最大行程
```

控制-1〈往复〉

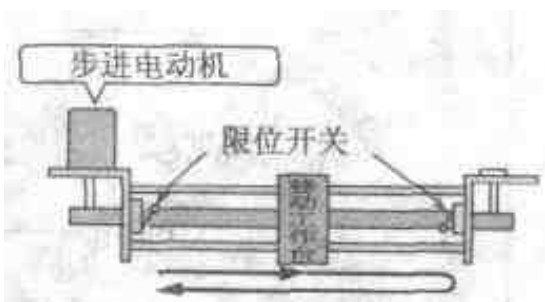


图 11 操作-1〈往复〉

(1) 动作

用来自脉冲发生电路的脉冲信号，以左端为起始点，以指定的次数使移动工作台在两端之间往复移行(参见图 11)。

① 将移动工作台设置在左端

让移动工作台往左运行，当确认左端限位开关动作时，就停止。

② 在屏幕上指定往返多少次。

③ 让移动工作台往右方移行，确认右端限位开关动作就停止。

④ 经一定时间后往左移行，确认左端限位开关动作，就停止。

⑤ 往返移行后，对左端限位开关每次增加 1 的动作进行计数，并与指定次数相比较，若一致就停止移行。

（2）程序

将运行程序输出→输入→比较→输出的程序，编写为一个函数程序。

函数syori（ ）程序

```
1 : syori(unsigned int od1,unsigned int hd)
2 : {
3 :     unsigned int ls;
4 :     outp(BPORT, od1);
5 :     do{
6 :         ls=inp(APORT)& hd;
7 :     }while (ls<=hd);
8 :     outp (BPORT,0x0);
9 : }
```

main 程序〈往复运行〉

```
#define APORT  0xd0      A口地址号
#define BPORT  0xd2      B口地址号
#define CPORT  0xd4      C口地址号
#define CWREG 0xd6       CW寄存器地址号
#define LLS    0x1       左端限位开关的输入数据
#define RLS    0x2       右端限位开关的输入数据
#define LIDO   0x6       向左移行输出数据
#define RIDO   0x4       向右移行输出数据
#define STP    0x0       停止输出数据
```

（通常标识名称都用大写字母）

标识名称的置换

```
syori(unsigned int hd,unsigned int odl)
{
        unsigned int ls;
        outp(BPORT,od 1);
        do{
                ls=inp(APORT)& hd;
        }while(ls<=hd);
        outp(BPORT,0x0);
}
main( )
{
        unsigned int nn,n;
        outp(CWREG,0x90);
        syori(LLS,SIDO);
        printf(¥x1B[2J);
        printf(" 往返多少次？ n=" );
        scanf("%d",&n);
        nn=0
        do{
                syori(RLS,RIDO);
                syori(LLS,LIDO);
                nn=nn+1;
        }while(nn!=n);
}
```

处理函数

输出移行的信号
①输出判断端部限位开关的信号
②在限位开关动作之前，反复进行输入判断
③若限位开关动作则停止输出信号

输入输出口的用法设置

将工作台设置在左端

清屏

用键盘输入

向右移行

向左移行

往返次数计数器



函数 syori()程序说明

形参如下：

odl：移行输出数据。

hd：限位开关输入数据。

4. 输出转向与脉冲发生的启动信号。

6. 将输入与限位开关的动作信号进行逻辑与(AND)运算，提取限位开关动作位的数据。

7. 在其数据与限位开关动作信号相等之前，反复执行上述 6. 的内容。

8. 输出停止数据。

控制-2(起始点设置)

编制将左端设置的起始点的函数程序[genten()]。脉冲由个人计算机输出。

(1) 动作

① 将移动工作台从左往右的设置

一边校验右端限位开关是否动作，一边输出向右方向移行的信号与一定数量的移行脉冲数。若右端限位开关动作，则停止移行。

② 向左移行

输出向左方向移行的脉冲，并向左移行。

若左端限位开关动作，则停止移行。

③ 起始点设置

在左端限位开关动作之前的位置，即发一个脉冲往右方向移行，左端限位开关也不会动作的位置，作为起始点(参见图 12)。

输出数据与①相同，加大脉冲间隔，缓慢移行。当输入数据达到上述①的话，就往相反方向。

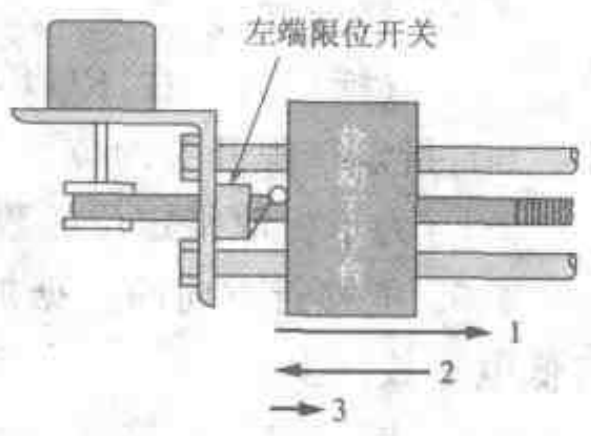


图 12　操作-2(起始点设置)

(2) 程序

编制输出脉冲，若限位开关动作，就停止的 PULS 函数程序。

函数〈puls(plsu,plhb,hoko,bit,limit)〉

```
1  puls(unsigned int plsu, unsigned int plhb, unsigned
   int hoko, unsigned int bit, unsigned int limit)
2  {
3      unsigned int t, h, ls, odt1, odt2;
4      odt1=hoko|PLSH;
5      odt2=hoko|PLSL;
6      for(t= =1;t<=plsu;t++)
7      {
8          outp(BPORT, odt1);
9          for(h= =1;h<=plhb;h++);
10         outp(BPORT, odt2);
11         ls=inp(APORT)&bit;
12         if(ls= =limit)
13         {
14             break;
15         }
16     }
17 }
```

程序说明－1

形参设置如下：

plsu：脉冲数（移行量换算的脉冲数）。

plhb：脉冲宽度（1 个脉冲高电平输出的时间）。

脉冲宽度越小，移行速度越快。

hoko：移行方向

（标识名　右：RGT，左：LET）。

bit：指定行程开关的位。

（为了提取指定位的数据）。

4、5. 把移行方向的数据加到（逻辑或（OR）运算）脉冲电压的高电平数据与低电平数据上。

6. 以指定的脉冲数，从 8～15 反复执行。

8. 输出脉冲电压的高电平状态。

9. 以脉冲宽度的数据，即保持时间，来保持高电平状态。

10. 输出脉冲电压的低电平状态。

11. 输入 A 口的数据，为了提取指定行程开关的数据，进行按位逻辑与（AND）运算。

12～15. 限位开关动作的话，跳出循环，结束函数程序。

〈起始点设置的函数程序〉

```
1  genten()
2  {
3      puls (100,50,RGT,RLS,RLS);
4      for (t= =1;t<=1000;t++);
5      puls (2000,50,LFT,LLS,LLS);
6      for (t= =1;t<=1000;t++);
7      puls (100,200,RGT,RLS,0x0);
8      outp (BPORT,0x10);
9      printf (¥x1B [2J);
10     printf ("《起始点设置结束》");
11 }
```

程序说明　2

1. 无形参。

3. 调用函数 puls()，从现在位置向右移行 100 个脉冲的距离，右端限位开关-动作，就结束函数程序，停止运行。

4. 一定时间后停止。

5. 调用 puls()，设置现在位置向左移行 2000 个脉冲的距离，左端限位开关-动作，就结束函数程序，停止运行。

6. 一定时间后停止。

7. 调用函数 puls()，设置从现在位置向左移行 100 个脉冲的距离，若左端的限位开关不动作，也结束函数程序，停止运行，以此位置作为起始点。

8. 起始点设置完毕，指示灯亮。

9. 清除屏幕。

11. 屏幕上显示：《起始点设置完毕》。

控制-3

以起始点(左端)为起点，按指定方向与距离移动。

(1) 动作

① 设置移动工作台的起始点。

② 在屏幕上输入移动方向与距离。

③ 当按下启动按钮，则移动到指定的位置(参见图 13)。

④ 反复执行②～③。

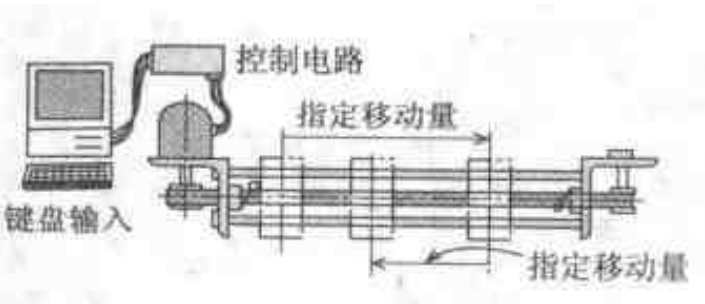


图 13　操作-3〈指定移动〉

(2) 程序

· 略去标识名称的定义与函数程序。

• 起始点设置，使用前述的已定义函数。genten()与函数 puls()。移动的执行使用函数 puls()。

• 从屏幕输入到移动执行，用 for(;;)语句无条件循环，执行完毕后，再执行屏幕上的问题，不再执行的场合，跳出循环圈，结束程序。

主要程序〈任意距离的移动〉

```
1  main()
2  {
3      outp(CWREG,0x90);          口使用方法的设置
4      genten();                  设置起始点
5      for(;;)
6      {
7          printf("¥x1B[2J");     清屏
8          printf(" 移动方向(向右+，向左-)与移动距离(mm)=" )
9          scanf("%d",&ido);      将键入的移动方向与距离的输入数据代入变量id
11         gokei+=ido;            累计移动距离
12         if(gokei<0 || gokei>MAXL)   判断移动范围
13         {
14             printf("<<超出移动范围，请再次修正>>")
15             continue;          返回for(;;)语句的开头
16         }
17         pls=ido/0.2;           将移动距离换成脉冲数
18         printf("<<是否可以?(Y/N)>>");
19         scanf("%s",&y);        了解移动的执行
20         if(y='n' || y='N')     将键入数据代入变量y
21         {
22             continue;          再执行时(输入N)返回for(;;)语句的开头
23         }
24         if(ido>=0)
25         {
26             puise(pls,50,RGT,RLS,RLS);   指定移动量为“＋”(省略＋)
27         }
28         else if(ido<=0)        进行右扫描处理
29         {
30             pulse(pls,50,LFT,LLS,LLS);   若指定移动量为“－”(省略－)
31         }
32         printf("<<是否继续?(Y/N)>>");     进行左扫描处理
33         scanf("%s",&cx);
34         if(x='n' || x='N')     判断是否要返回到程序开头，继续执行
35         {
36             break;             N的情况下
37         }
38     }
39 }                              跳出无条件循环for(;;)语句，结束程序
```

4 铁路模型的控制

用C语言来编制简单的铁路模型控制程序。

1 铁路模型

轨路配置如图 2 所示。

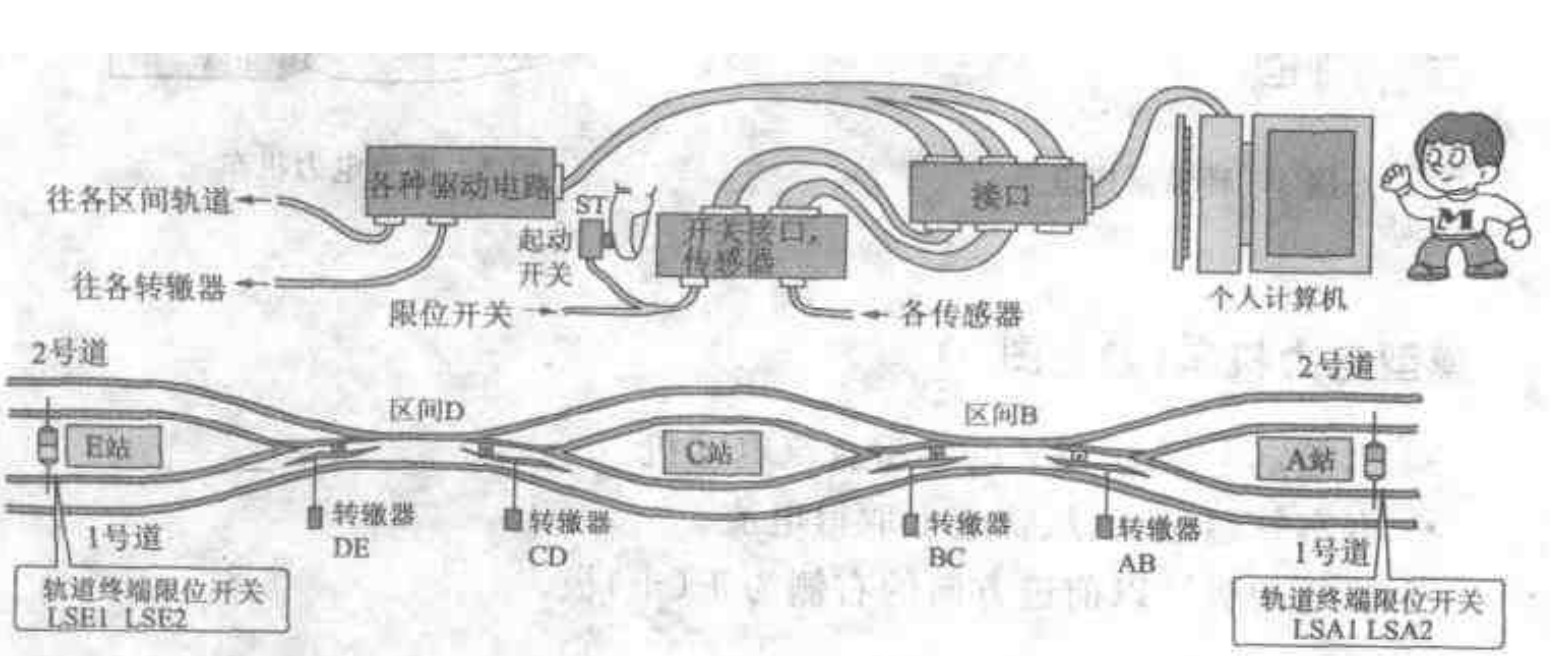


图 2 铁路模型构造图

轨 道

· 将 A 站、C 站、E 站与单轨区间 B、D 连接，各站均设置双轨。双轨与单轨用转辙器 AB、BC、CD、DE 连接。

· 轨道共分 A1、A2、B、C1、C2、D、E1、E2 八个区间，各自独立供电。转辙器 AB 与 BC 和区间 B，转辙器 CD 与 DE 和区间 D 的极性相同。

· 在 A 站与 E 站的轨道终端，装有防止通过的限位开关，并将信号输入计算机。

· 电力机车检测用传感器是以检测轨道上流过电流的方法，装在各区间的电源电路上。

转辙器

· 转辙器用于选择轨道，摆动转辙器的轨道，就可与运行轨道同路的轨道相连接(参见图 1)。

· 转辙器通过直流电动机与螺杆机构将旋转运动转化为直线运动。在转辙器的摆动终端，设置检测用的限位开关。

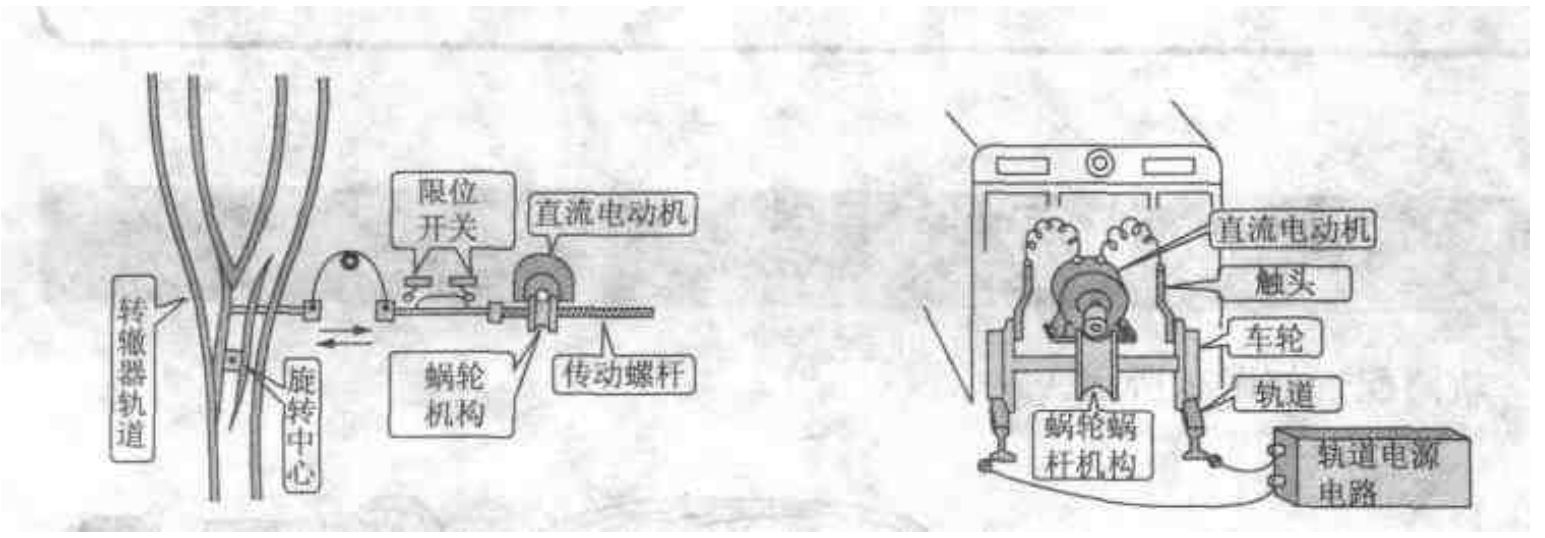


图 1　转辙器构造图　　图 3　模型电力机车

模型电力机车(参见图 3)

· 用额定电压为 12V 的小型直流电动机。
· 有车轮通过，则从轨道上取得电流。
· 轨道的极性以前进方向的右侧为正(+)极。

Ⅱ　主电路

轨道电源电路

轨道电源的供电，是通过两个作为“电源开关”与“极性变换”用的电磁继电器的接点连接到轨道上的。电磁继电器由来自计算机的信号并通过晶体管控制。各区间之间的供电线路连接如图 4 所示。

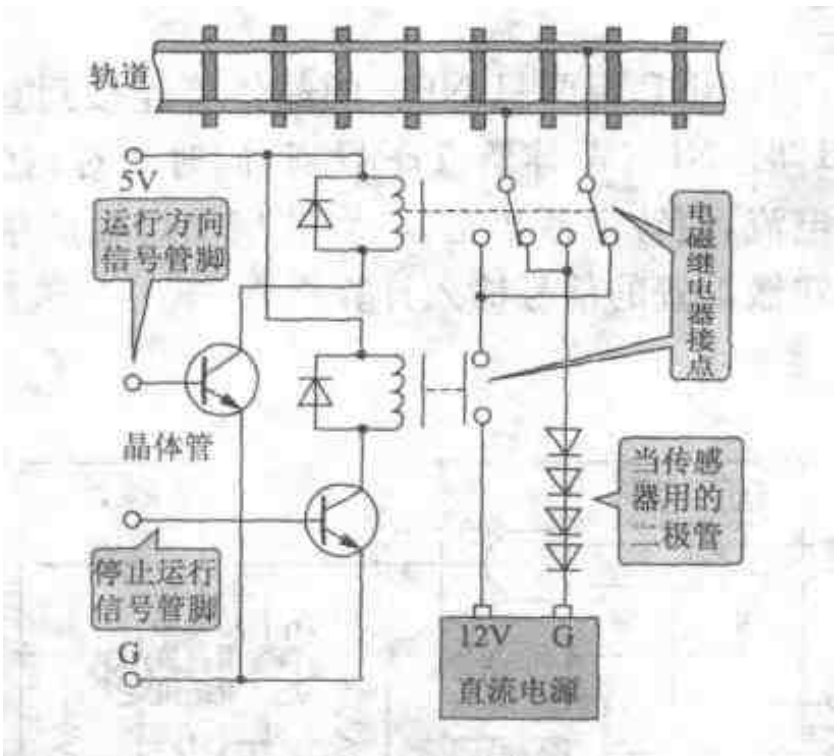


图 4　轨道电源电路

电力机车检测传感器

电力机车检测传感器是检测电力机车是否进入该区间轨道的传感器，电力机车在轨道上运行时，通过检测流经机车的电流来检测电力机车（参见图 5）。

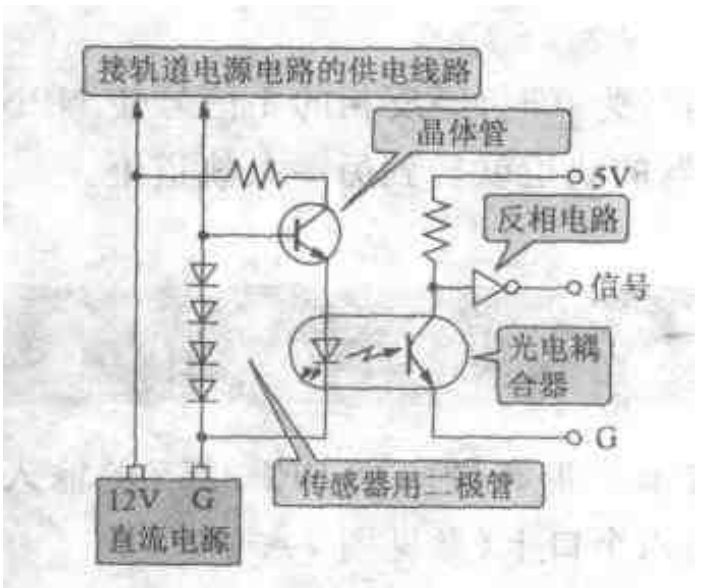


图 5　电力机车检测传感器

四个二极管串联连接在电源电路上。将流过电流产生的电位差，作为使晶体管、光电耦合器工作的信号，将这一信号输入计算机，作为检测电力机车的信号。

转辙器驱动电路

如图 6 所示，两只 PNP 与两只 NPN 的晶体管连接到直流电动机，当来自计算机的信号为“1”时，NPN 晶体管 Tr_4 导通，同时相对侧的 PNP 晶体管 Tr_1 导通，电动机流入电流，使转辙器轨道摆动，若摆动终端的限位开关动作，电动机就停止，将完成转辙设置的信号输入计算机。

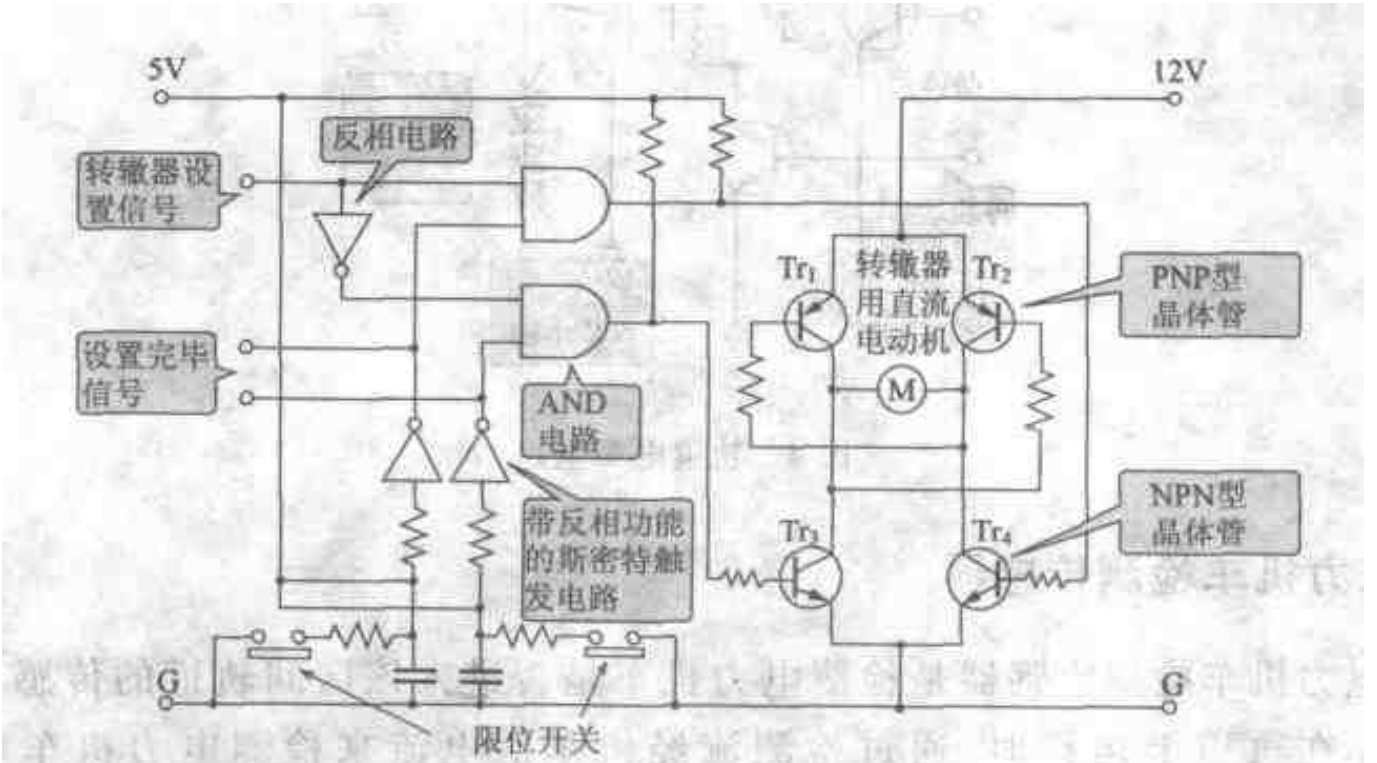


图 6　转辙器驱动电路

若来自计算机的信号为 0 时，经反相的 1 信号使 NPN 晶体管 Tr_3 到 PNP 晶体管 Tr_2 导通，转辙器的轨道就转到另一条轨道上。

Ⅲ　与个人计算机的连接

个人计算机的扩展槽上带有带两个 PPI 8255 的输入输出接口插接件，将各电路连接在插接件的六个口上(参见图 7)。

① 口的设定

使用模式为模式 0，将 8255－1 的三个口设定为输入口，8255－2 的三个口设定为输出口。

表示这一设置的 CW 数据如下，用程序将数据输出到各 8255 片的 CW 寄存器中。

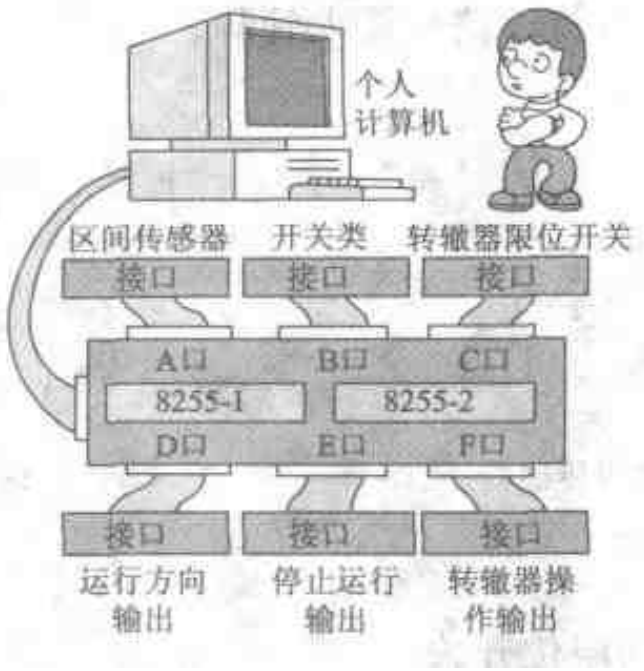


图 7 与个人计算机的连接

8255－1……全部口均用于输入。

$(1001\ 1011)_2=(9B)_{16}$

CW 寄存器地址号→$(D6)_{16}$

8255－2……全部口均用于输出。

$(1000\ 0000)_2=(80)_{16}$

CW 寄存器地址号→$(DE)_{16}$

② 口的地址号(地址)

8255－1 的 A、B、C 口，8255－2 的 C、D、F 口的地址设置如下。

A 口→$(D\ 0)_{16}$

B 口→$(D\ 2)_{16}$

C 口→$(D\ 4)_{16}$

D 口→$(D\ 8)_{16}$

E 口→$(D\ A)_{16}$

F 口→$(D\ C)_{16}$

③ 电力机车检测传感器输入

如图 8 所示，有 8 个区间，将这些电力机车的检测传感器连接到 A 口的 8 位上。

电力机车一进入区间的轨道，1 信号就被输入到输入口的管脚；没有电力机车时，0 信号被输入到输入口的管脚。

④ 开关输入

如图 9 所示，将启动开关 ST、停止开关 STP 与各轨道终端的限位开关(LSA1、LSA2、LSE1、LSE2)连接到 B 口的各位上。

若开关接通(ON)，就将 1 信号输入管脚。

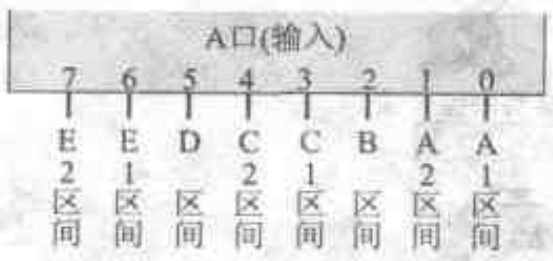


图 8　电力机车的检测传感器



图 9　各种开关

⑤ 转辙器设置用的限位开关

如图 10 所示，将转辙器驱动装置摆动两端限位开关的信号引脚，连接到 C 口上。

转辙器轨道驱动用的螺杆轴上，装有端部限位开关，用此开关动作的信号，作为确认转辙器完成设置动作的信号。

⑥ 电力机车运行方向输出

如图 11 所示，将各区间轨道电源电路的运行方向信号管脚接到 D 口上。

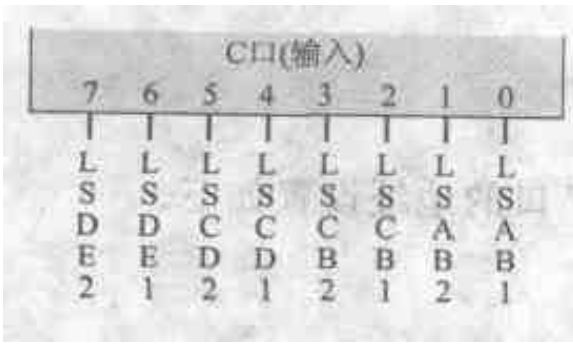


图 10　转辙器设置用的限位开关



图 11　运行方向输出

从 A 站往 E 站前进方向(往下)输出 0 信号，从 E 站往 A 站前进方向，输出 1 信号，以此来设置电磁继电器。

⑦ 电力机车停止运行输出

如图 12 所示，将各区间的电力机车电源电路停止运行信号管脚与 E 口连接。

电力机车运行输出 1 信号，停止运行输出 0 信号。

⑧ 转辙器的设置输出

如图 13 所示，转辙器的驱动电路设置信号管脚连接到 F 口上。

设置往 1 号轨道，输出 0 信号；往 2 号轨道，输出 1 信号。



图 12　停止运行输出



图 13　转辙器的设置输出

Ⅳ 控制程序

用 C 语言，编制简单电力机车运行控制的程序。

程序-1(单轨运行)

■动作

一按 ST(启动)开关，就依 A 站 1 号轨道→C 站 2 号轨道→E 站 1 号轨道的路线运行(参见图 14)。

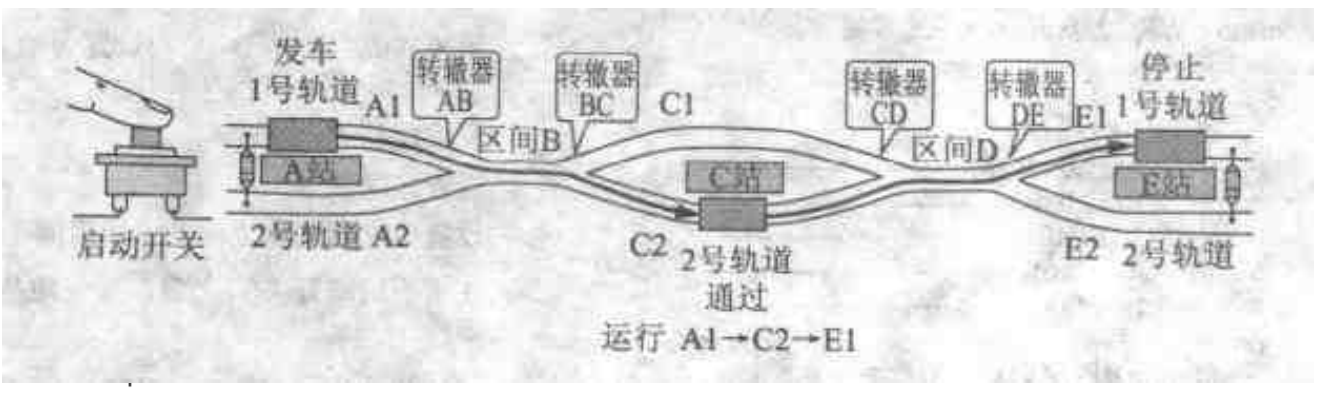


图 14　操作-1(单轨运行)

① 转辙器设置

将转辙器 AB 与 DE 连接在 1 号轨道，将转辙器 BC 与 CD 连接在 2 号轨道(F 口的第 0 位与第 3 位输出 0 信号，第 1 位与第 2 位输出 1 信号)。

输出数据(F 口)：$(0000\ 0110)_2=(6)_{16}$

将转辙器设置在规定位置，并用 C 口的限位开关输入信号来确认(确认在第 0、3、5、6 位输入了 1 信号)。

输入数据(C 口)：$(0100\ 1001)_2=(69)_{16}$

② 运行方向设置

设置轨道区间 A1、B、C2、D、E1 的 A→E 运行方向(在 D 各位上输出 0 信号)。

输出数据(D 口):$(0000\ 0000)_2=(0)_{16}$

③ 开始

按下 ST(启动)开关,电力机车从 A 站的 1 号轨道出发,从 A→E 方向运行(B 口的第 0 位输入 1 信号,则 E 口的第 0、2、4、5、6 位就输出 1 信号)。

输入数据(B 口):$(0000\ 0001)_2=(1)_{16}$

输出数据(E 口):$(0111\ 0101)_2=(75)_{16}$

④ 停止

当确认电力机车进入了 E 站的 1 号轨道后,就让车停在站的中间(在 A 口第 6 位输入 1 信号,则 E 口的所有位均输出 0 信号)。

输入数据(A 口):$(0100\ 0000)_2=(40)_{16}$

输出数据(E 口):$(0000\ 0000)_2=(0)_{16}$

■ 从 A 站到 E 站的单轨运行程序

```
#define APORT  0xd0 · · · A口
#define BPORT  0xd2 · · · B口
#define CPORT  0xd4 · · · C口
#define DPORT  0xd8 · · · D口
#define EPORT  0xdA · · · E口
#define FPORT  0xdC · · · F口
#define CWRG1 0xd6 · · · CW寄存器1
#define CWRG2 0xdE · · · CW寄存器2
main()
{
    unsigned int t,st,ds;
    outp(CWRG1,0x9b);
    outp(CWRG2,0x80);
    outp(DPORT,0x0);
    outp(FPORT,0x6);
    for(t=0;t=>2000;t++);
    do{
        st=inp(BPORT)& 0x1;
    }while(st!=0x1);
    outp(EPORT,0x75);
    do{
        ds=inp(APORT)&0x40;
    }while(ds!=0x40);
    for(t=0;t=>3000;t++);
    outp(EPORT,0x0);
}
```

定义标识名称

用模式0，将A、B、C口设置为输入口，将D、E、F口设置为输出口

设置运行方向

设置转辙器

设置转辙器到位之前的一段时间

在按下启动开关ST之前，反复地从B口输入

A站1号轨道→B区间→C站2号轨道→D区间→E站1号轨道的运行

在E站1号轨道的检测传感器，检测到电力机车之前，反复地从A口输入

电力机车到达中间之前运行一段时间

使电力机车停止

程序-2(两台电力机车的交会)

编制如下的两台电力机车的控制程序(参见图 15)。

■ 动作

按下启动开关 ST,电力机车Ⅰ从 A 站的 1 号轨道发车,进入 C 站的 2 号轨道就停止。改变转辙器与运行方向的设置,电力机车Ⅱ从 E 站的 2 号轨道发车,通过 C 站的 1 号轨道,到 A 站的 2 号轨道停止。

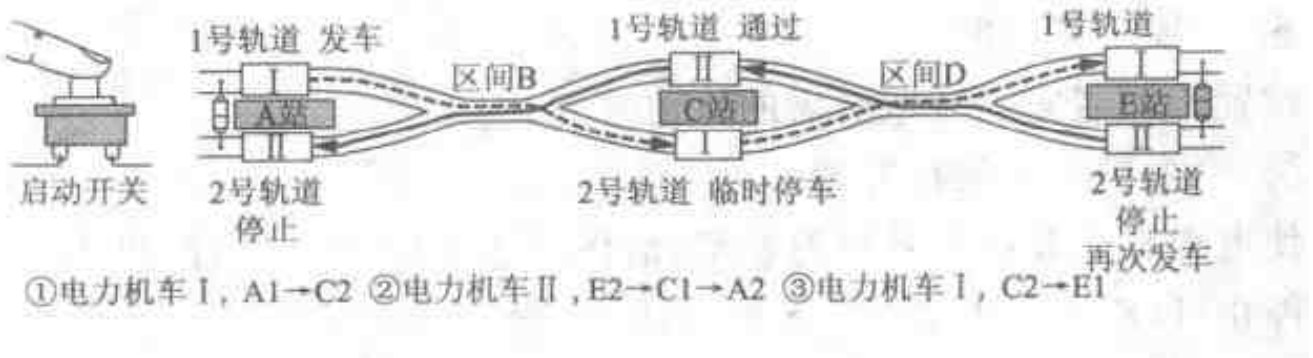


图 15 操作-2(更换输入程序)

再改变转辙器与运行方向的设置，使Ⅰ号电力机车向E站发车。当Ⅰ号电力机车驶进E站后停止。

① 转辙器设置

将转辙器AB、CD设置在1号轨道，将转辙器BC、DE设置在2号轨道(F口的第0位、第2位输出0信号，第1位第3位输出1信号)。

输出数据(F口)：$(0000\ 1010)_2=(A)_{16}$

② 运行方向设置

将区间A1、B、C2设置为向下运行，将区间C1、D、E2设置为向上运行(D口的第3位、第5位、第7位输出1信号)。

输出数据(D口)：$(1010\ 1000)_2=(A8)_{16}$

③ 电力机车Ⅰ运行开始

按下启动开关ST，则电力机车Ⅰ从区间A1→B→C2运行(B口的第0位输入1信号，则E口的第4位、第2位、第0位就输出1信号)。

输入数据(B口)：$(0000\ 0001)_2=(1)_{16}$

输出数据(E口)：$(0001\ 0101)_2=(15)_{16}$

④ 电力机车Ⅰ的C站停车

若检测到电力机车Ⅰ进入C站的2号轨道后，就使电力机车在站中间附近停车(A口的第4位若输入0信号，则E口的全部位均输出0信号)。

输入数据(A口)：$(0001\ 0000)_2=(10)_{16}$

输出数据(E口)：$(0000\ 0000)_2=(0)_{16}$

⑤ 运行方向、转辙器的再次设置

设置区间A2、B、C1、D、E2为向上运行(D口的第1、2、4、5、7位输出1信号)。

输出数据(D口)：$(1011\ 0110)_2=(B6)_{16}$

将转辙器AB、DE设置在2号轨道，将转辙器BC、CD设置在1号轨道(F口的第0位，第3位输出1信号，第2位、第1位输出0信号)。

输出数据(F口)：$(0000\ 1001)=(9)_{16}$

⑥ 一段停留时间

设置转辙器完成转辙所需的停留时间。

⑦ 电力机车Ⅱ运行开始

使电力机车Ⅱ从 2 号轨道发车，沿区间 D→C1→B 运行，进入 A 站 2 号轨道后停止(D 口的第 1、2、3、5、7 位输出 1 信号)。

输出数据(D 口)：$(1010\ 1110)_2=(AE)_{16}$

⑧ 电力机车Ⅱ A 站停车

若检测到电力机车Ⅱ进入了 A 站的 2 号轨道，就在站中间附近停止(若确认 A 口的第 1 位输入了 1 信号)，则一段时间后，全部 D 口的位均输出 0 信号)。

输入数据(A 口)：$(0000\ 0010)_2=(2)_{16}$

输出数据(D 口)：$(0000\ 0000)_2=(0)_{16}$

⑨ 运行方向转辙器的设置

设置沿区间 C2、D、E1 向下运行，沿区间 A2、B、C1 向上运行(D 口的第 1、2、3 位输出 1 信号)。

输出数据(D 口)：$(0000\ 1110)_2=(E)_{16}$

将转辙器 AB、CD 设置在 2 号轨道，将转辙器 BC、DE 设置在 1 号轨道(F 口的第 0 位、第 2 位输出 1 信号，其他各位输出 0 信号)。

输出数据(F 口)：$(0000\ 0101)_2=(5)_{16}$

⑩ 一段时间的停止

用一段时间让转辙器完全设置完毕。

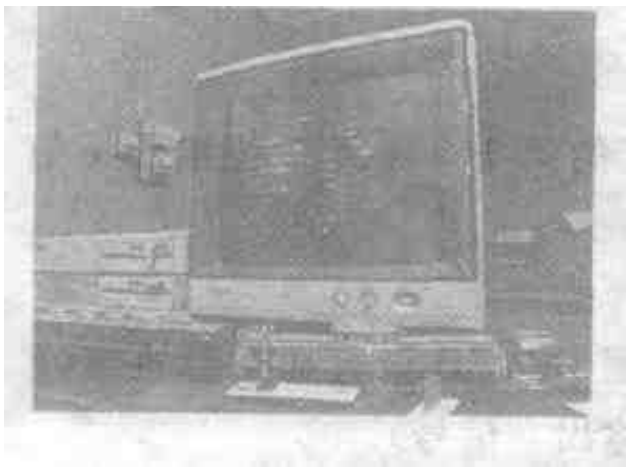
⑪ 再次发车

将电力机车Ⅰ沿区间 C2→D→E1(D 口的第 4、5、6 位输出 1 信号)。

输出数据(D 口)：$(0111\ 0000)_2=(70)_{16}$

⑫ E 站停止

若确定电力机车Ⅰ进入了 E 站 1 号轨道，就在站中间附近停止(A 口的第 6 位输入 1 信号的话，一段时间后，D 口所有的位输出 0 信号)。



输入数据(A 口)：$(0100\ 0000)_2=(40)_{16}$

输出数据(D 口)：$(0000\ 0000)_2=(0)_{16}$

⑬ 再次执行

在屏幕上问是否再次执行，并作出决定。

右图是电力机车模型与控制计算机的照片。

⑭ 程序。

编制判断运行方向，转辙器设置的函数与输出函数等程序。

■ 两台电力机车的交会　　更换程序中省略了标识名称的定义

```
1  settei(unsigned int hokodt,unsigned int pontdt)
2  {
3      outp(DPORT,hokodt);
4      outp(FPORT,pontdt);
5      for(t=0;t=>2000;t++);
6  }
7  syori(unsigned int inpt,unsigned int otpt,unsigned int indt,unsigned int otdt,int unsigned time)
8  {
9      unsigned int ds,t;
10     do{
11         ds=inp(inpt)& indt,
12     }while(ds!=indt);
13     for(t=0;t->time;t++),
14     outp(otpt,otdt);
15 }
16 main()
17 {
18     unsigned int t,a;
19     outp(CWRG1,0x9b);
20     outp(CWRG2,0x80);
21     do{
22         settei(0xa8,0xa)
23         syori(BPORT,EPORT,0x1,0x15,0);
24         syori(APORT,EPORT,0x10,0x0,5000),
25         settei(0xae,0x9)
26         outp(EPORT,0xae);
27         syori(APORT,EPORT,0x1,0x0,3000);
28         settei(0xa8,0xa)
29         outp(EPORT,0x70);
30         syori(APORT,EPORT,0x40,0x0,5000);
31         printf("¥x1B[2J");
32         printf("结束吗？(是/否)"),
33         a=getchar();
34     }while(a!='y');
35 }
```

函数settei程序
<转辙器与运行方向的设置>



函数syori程序
<根据设置输入进行输出处理>

设置A、B、C各口为输入口

设置D、E、F各口为输出口

运行方向与转辙器的设置

执行syori()函数，若按下启动开关ST，则电力机车向下运行

执行syori()函数，若电力机车Ⅰ驶入了C站的2号轨道的话，就停止

设置运行方向与转辙器

使电力机车Ⅱ沿E2→D→C1→B→A2区间运行

若电力机车Ⅱ进入了A站的2号轨道的话，就停止

设置运行方向与转辙器

使电力机车Ⅰ沿C2→D→E1区间运行

若电力机车Ⅰ进入了E站的1号轨道的话，就停止

5 中断控制的设计

中断的定义是在程序执行过程中、要执行临时插入的一项工作的程序。例如，当检测到机械的动作异常时，就执行紧急停止等在控制中常用的程序。计算机中有两种中断方法，使用中断请求端子的硬件方法与在程序中编制的软件方法。

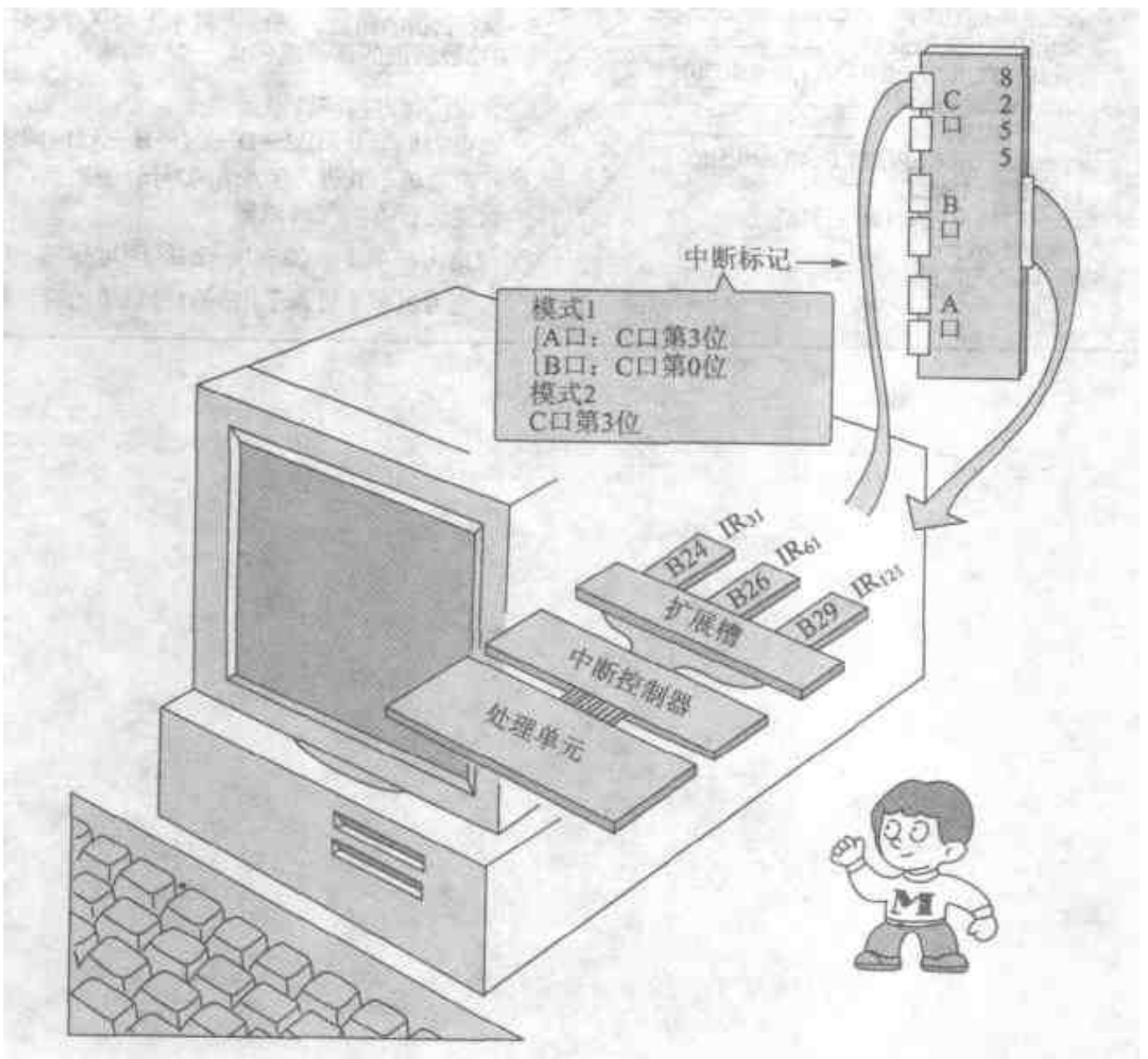


图 1　8255 与中断

Ⅰ 硬件的方法

PC9801 的扩展槽中，有 7 个管脚可用于中断输入，其中可无条件供用户使用的有 3 个，要与有关管脚连接使用。如显示应答中断请求的中断响应信号管脚，与中断有关的各种接口的管脚等。在个人计算机的内部，扩展槽的中断请求信号，是通过中断控制器的处理单元的中断请求信号管脚输入的。

用 8255 接口来设置中断请求输出管脚，有模式 1 与模式 2 两种中断模式。

若处理单元让中断请求信号管脚一有输入，就立即执行事先设置的中断处理程序，其处理速度要比软件处理的中断方法快，在控制中用得较多。

Ⅱ 软件方法

在控制程序中设置经常地检测相当于中断请求信号（例如紧急停止信号等）的程序，检测到这种输入信号就执行中断处理程序。

〈例〉步进电动机驱动程序中紧急停止（参见 7.7 节）。在前面步进电动机驱动程序中，编一段加入检测、处理紧急停止信号的程序。在发生脉冲的函数程序中加入信号的检测程序。假定在 A 口的第 7 位中设置紧急停止信号。

① 脉冲发生函数 pulse(n, h)的修改

```
34  pulse (n,h)
35   {
36      unsigned int i,t,od,d,wari;
37      for (i=1;i<=n;i++)
38      {
39        for (d=0x0;d<=0x1;d++)
40        {
41          od=ho+d;
42          outp(0xd2,od);
43          wari=inp (0xd0)&0x80;
44          if (wari==0x80)
45              {
46                  outp(0xd2,0x0);
47                  goto break;
48              {
49        }
50      }
51  break;;
52  }
```



② 程序说明

39～49. 将输出数据代入变量 od 中，与相应的旋转方向数据 0x0 或 0x1 交替输出。

43～48. 中断判断与处理。

43. A 口的数据输入变量 wari 中，用 0x80 进行按位逻辑与(AND)运算，输出第 7 位的数据。

44. 判断变量 wari 的数据。

46. 是 0x80 的话，B 口输出 0x0。

47. 用第 51 行的标记 Break 跳出循环，脉冲发生函数程序结束。

③ 脉冲的周期

中断判断与处理程序的处理时间，对脉冲周期有影响，因为由程序来计算周期有困难，所以要在执行程序的过程中，用示波器来观察波形的周期。

周期的长短，可在第 42 行的输出命令后面加入 for 语句，用循环次数来调整。

练习题

1. 中断是什么意思？
2. 中断的方法有几种？
3. 用处理单元的中断请求管脚输入中断的方法，有何优点？
4. 用 8255 输入输出接口中断的场合，可用几种模式？

解答

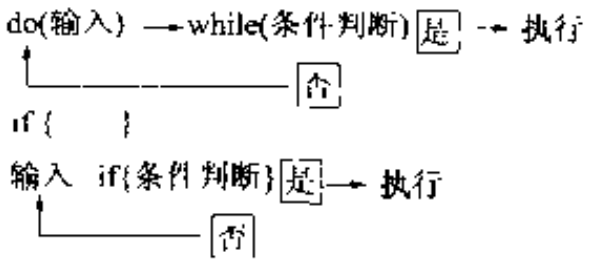
1. 在通常的程序执行过程中，一加入某个信号，就中止所执行的程序，而执行中断程序，这就叫中断。
2. 有硬件方法与软件方法。
3. 用硬件方法的处理速度快，当中断请求信号的种类多的情况下，使用方便。
4. 可用模式 1 与模式 2 两种

挑战题

这一章出的是控制程序的问题。

① 第一节的"用气动执行机构传送装置"的控制程序中，若将 do while 语句之处改写为 IF 语句，如何实现？

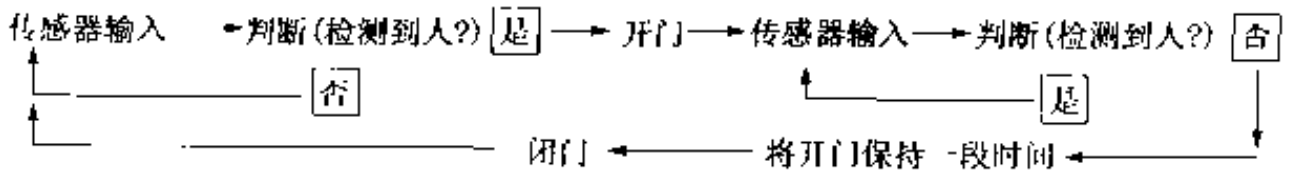
（提示）do-while





② 第 2 节"自动门控制"中，先编制一个不考虑模型的自动门安全性的简单操作控制程序。然后再考虑安全性，方便性等进行修正，试编制自己的控制程序

（提示） 简单控制



门的开、闭动作(电动机驱动门机构，开、闭门限位开关接通(ON，电动机停止)

③ 在第 3 节"步进电动机传送装置"中，设计一个用于移行时的移动量存储程序。另外设计一个要存储多个移动量，能将其操作再现的程序。

（提示）

1） 用脉冲发生电路的脉冲信号来驱动步进电动机，同时将该脉冲信号输入计算机。

2） 累计此脉冲数，存到配置的变量中。

3） 操作再现，将所存储的脉冲数从计算机输出到步进电动机驱动电路使其移行。

④ 第 4 行"铁路模型的控制"中，设计一个一台电力机车在 A、E 站之间往返的程序。或有两台电力机车分别在 A 站与 E 站同时发车，在 D 站停止，经过转辙器交会后，各自向对方车站出发的程序。

（提示）

1） 有必要考虑转辙器的设置。

2） 两台电力机车同时进入各站是没有限制的，因此有必要特别注意进站的判断，电力机车停止的延续时间。

A 答 案

①～④的答案略（请详细阅读有关章节）。

第9章

相扑机器人的制作技巧



本章的目的

本章叙述"相扑机器人的制作技巧"。

尽管胜负被说成是运气，但肯定是胜方得意。通过辛勤劳动与智慧制作出来的相扑机器人，犹如"自己的分身在比赛"。

按下启动按钮，在等待5秒钟的时间内，尽管至今为止经过了无数次的反复检查，我总是提心吊胆，稍微夸张地说，心情简直像花数百亿日元发射人造卫星到了倒计时时的那样紧张。

"干吧!加油!"，从那些不是裁判的裁判员发出的叫喊声开始，就开始了胜负的角逐，也传来了无情的"胜利了!"的声音。

"相扑机器人"的趣味，就是"自制"所带来的乐趣。

别人可能把相扑机器人看成是"会动的快餐盒"，然而它在我眼中却是珍贵无比的珠宝盒。

别的学校的同事们所制作的相扑机器人，不也作为实践演习吗?不知不觉地想到，也许你的孩子们讲到过"从前做过相扑机器人呀"等话，我们就应迎接挑战，来精心制作机器人。

1 相扑机器人的定义



I 相扑机器人的诞生

相扑机器人这个术语，在日本关心机电一体化和计算机的人们中间，已深人人心了。所以，如普通人所知道的："啊，电视播放过有个赛场上，装在小车上的快餐盒似的机器人来回挤撞，可真是场面火爆的比赛呀！"

这次相扑机器人比赛，是1990年富士软件株式会社作为纪念公司创立一周年的项目来策划的。同年3月，就实施了第一届的比赛大会。这次大会对年轻的工程师、学习机电一体化的人或喜欢动手制作的人来说，是一边学习技术，一边展示所制作的作品（相扑机器人）的极好场所。其后，曾多次举行大会，使机器人相扑日益流行。

• 大学生机器人相扑比赛大会的举办

1993年在富山县，1994年在京都府，都举办了大学生机器人相扑比赛的全国（决赛）大会。

对于大学生来说，制作"相扑机器人"是一 边用自己的手，通过辛勤劳动学习先进技术，一边又参加了全国大学中的教材比赛。这些与夏天机器人足球比赛一样火爆，涌现了大学生埋头在焊锡烟气中制作相扑机器人的新热潮。

Ⅱ　相扑机器人的种类

本书中，将机器人相扑的相扑运动员本体称为“相扑机器人”。

相扑机器人分为智能型与遥控型两类，各自均有一款，取名为“干吧”。

智能型

相扑机器人的本身装有控制机器人动作用的计算机。

同时还用计算机汇总来自各种各样传感器的信息，来控制机器人的动作，与对方机器人战斗。

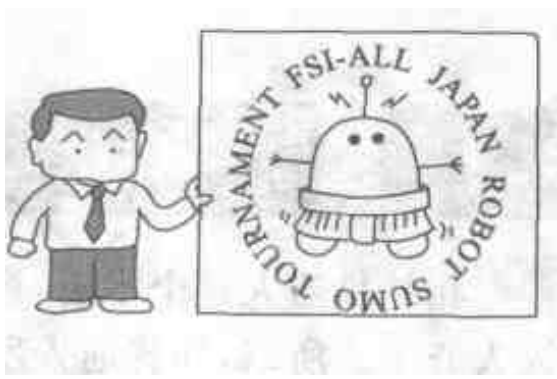
自身可以控制的相扑机器人，称为智能型相扑机器人。

所用计算机一般都用单片微机。

另外我想，使用身边现成的笔记本式计算机，以 C 语言控制的“用笔记本式计算机 C 语言的相扑机器人作品”，也是很有趣的挑战。

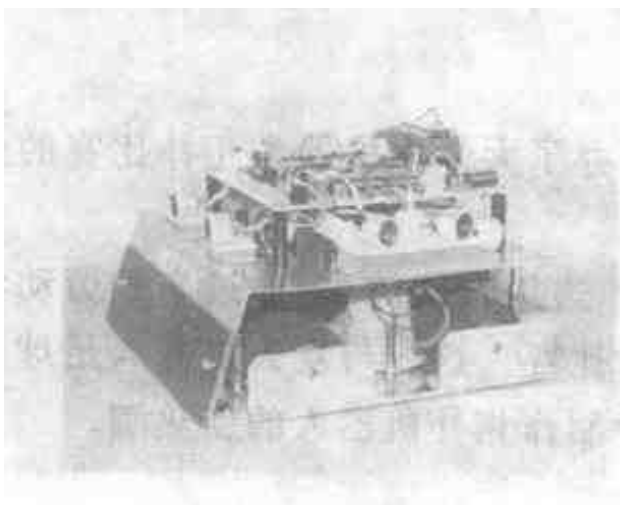
无线遥控型

通过无线控制操作（发射机），来控制机器人动作的相扑机器人，称为遥控型相扑机器人，它需要遥控操作的高技术。

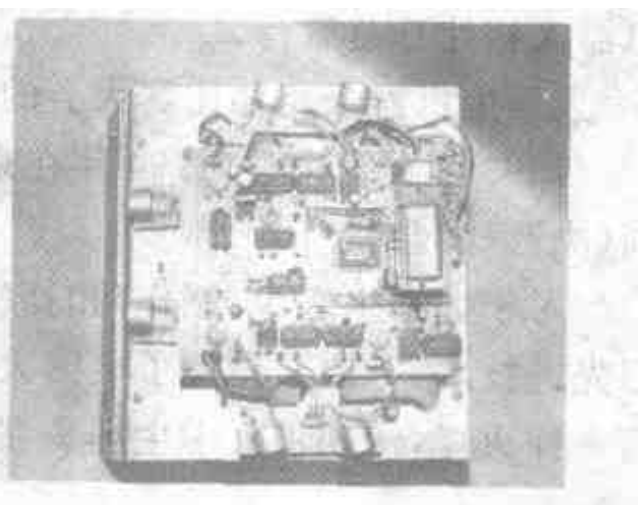


Ⅲ　相扑机器人的照片

这些照片都是在赛场中参加过比赛的一些相扑机器人。



四轴驱动式（智能型）



智能型 CPU 与传感器



用乒乓球拍橡胶自制的轮子(智能型)



两轴驱动式(智能型)

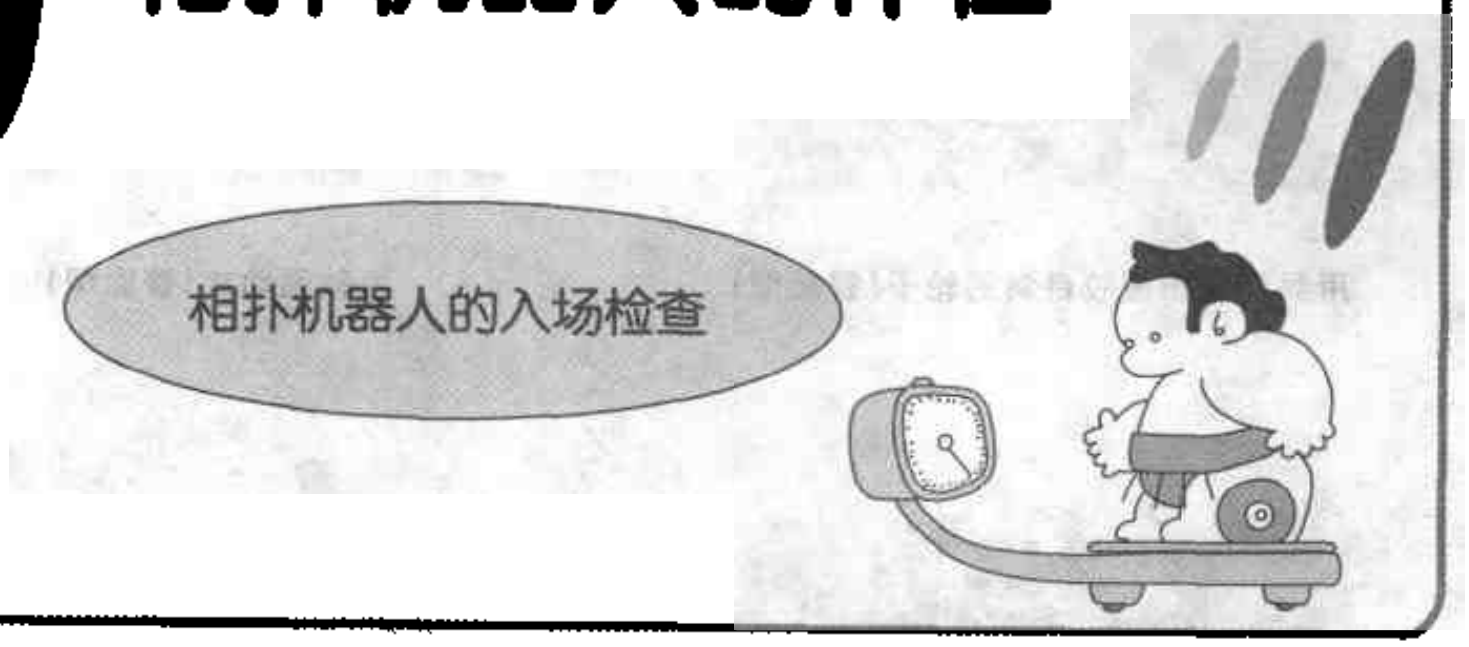


遥控型



带武器的遥控型

2 相扑机器人的体检



I 相扑机器人的体检(标准)

相扑运动员入场时要进行体格检查。因为相扑机器人要在同等条件下竞争智慧与技巧,所以要有图1、图2所示的“大小与重量的限制”(参见表1、表2)。

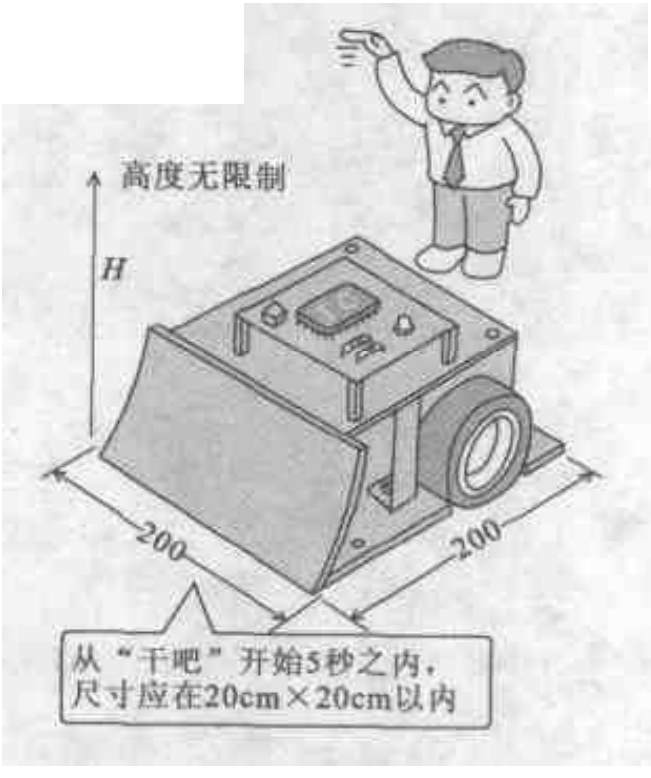


图1 相扑机器人的大小

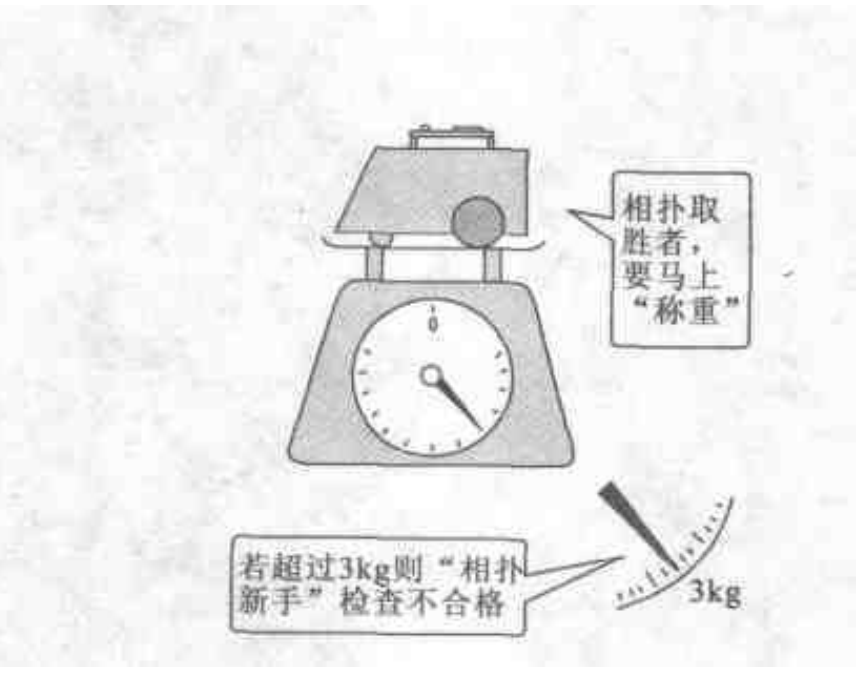


图2 相扑机器人的重量

表 1 相扑机器人的大小

根据标准大小	宽	20cm 以下
	深	20cm 以下
	高	无限制

但是，比赛开始后的相扑机器人的手腕或脚等本体部分与附属部分是允许伸缩的。如表 2 所示，智能型相扑机器人的本体，电动机的驱动电源，接插件、电缆、操作开关等附属物，所含的重量应当在 3000g 以内才算合格，超过则失去参赛资格。

表 2 相扑机器人的重量

重量	3000g(3kg)以内(包括附件)

要将对方的机器人推出去的力量，取决于电动机力矩的大小，此电动机的重量，要占到相扑机器人本身重量的 20%～30%。

因此推力的大小，受电动机重量的影响也很大。选多大的电动机，并且要兼顾该电动机驱动用 IC 能否买到等等，是要用点心思的。

另外，遥控型的场合，遥控器的重量是不包括在相扑机器人之内的。

Ⅱ 主要的决胜规则(参见图 3)

(一次比赛有三场，先胜两场者取胜。)

一场取胜

判定取胜

根据在赛场上动作之外的因素，来判断胜负。

其 他

根据对方违反相扑机器人的规定，决定胜负或抽签决定胜负。

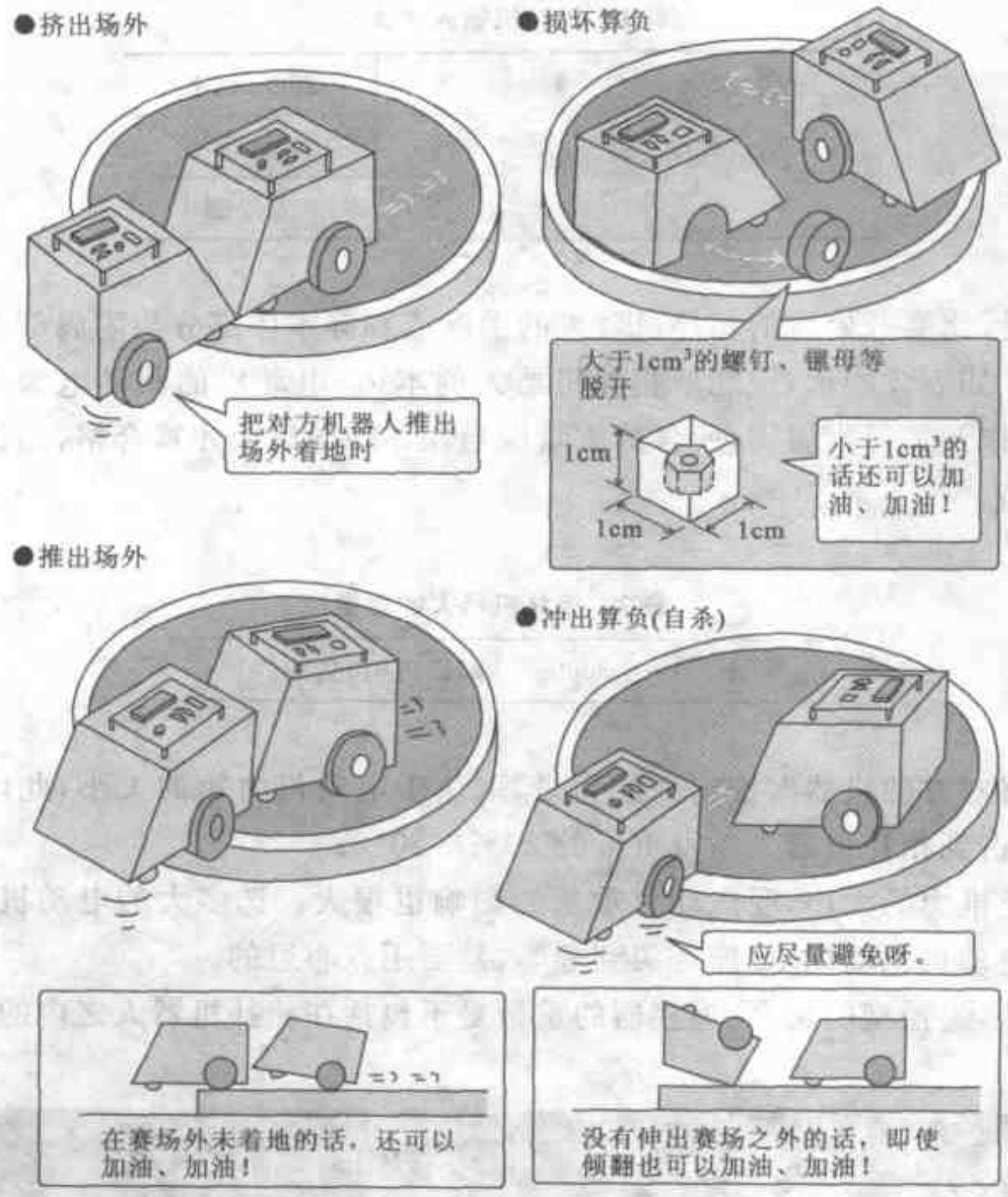

●挤出场外
把对方机器人推出场外着地时
●损坏算负
大于1cm³的螺钉、螺母等脱开
1cm
1cm
1cm
小于1cm³的话还可以加油、加油！
●推出场外
在赛场外未着地的话，还可以加油、加油！
●冲出算负(自杀)
应尽量避免呀。
没有伸出赛场之外的话，即使倾翻也可以加油、加油！

图 3　主要的决胜规则

3 相扑机器人的组装(1)

相扑机器人的主要形状



- **相扑机器人本体的构造**

相扑机器人的构造，其大小、重量均要在规定的范围之内，为了制作出强有力的机器人，制作者是要动一番脑筋的。这里介绍制作者们的一些主要技巧。

Ⅰ 相扑机器人的形状

楔块型相扑机器人(参见图 1)

一般相扑机器人的形状为两轴驱动型(两个电动机，即两轮驱动)。

1994 年，日本关东地区比赛大会所参加的相扑机器人，约有 70%是两轴驱动的楔块型机器人。其理由大致如下：

- 相扑机器人的稳定性好。与四轴驱动相比，两轴驱动型易于平衡地布置所有的零件(如图 1 所示，在楔块前端布置赛场检测传感器，中间布置电动机驱动电源，后面放置驱动电动机)。
- 易于钻入对方相扑机器人的下方。楔块形在比赛过程中，最适宜的取胜方式是“钻入”对方相扑机器人的下面。



图 1　楔块形相扑机器人

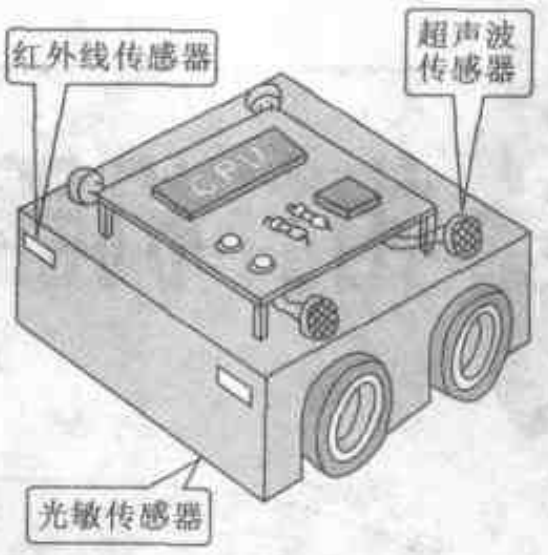


图2　快餐盒型(四轴驱动)相扑机器人

• 可装大功率的电动机。两轴驱动型有较大的空间，能装大功率电动机与驱动轮胎。

快餐盒型相扑机器人(参见图 2)

四轴驱动型大多就是所谓“快餐盒”型，与楔块型相比，在前端也装有 1～2cm 左右的楔块金属，当与对方相扑机器人碰撞时，是力图使对方抬起来的部件。

这是“相互推挤以力取胜”类型，四个电动机能产生可引以自豪的力量。

II　轮胎数(电动机数)

轮胎数就是电动机数，也就是驱动轴数，有 2 轴型、4 轴型和 6 轴型几种。

相扑机器人大部分都是 2 轴，4 轴型的。一般说来，电动机数量与力量成正比，但是与四轴的中型电动机相比，两轴的强力电动机要更有力量一些。

我想无论如何配置不同的驱动轴数，在电动机与减速比等各种条件的选择上，也会影响到胜负。

对两轴型来说，是把后轴作为电动机轴的，在前轮部位，则常在中央部位配置一个 12mm 的球形万向轮，或者在左右各配一个球形万向轮。

借用遥控模型用的车轮

在无线遥控专卖店可买到各种尺寸的车轮。将塑料表面涂漆加工，就可以当轮子用。购买时应当想好，是买铝制的，还是买塑料制造的。当然，铝制品的强度要好得多。

自制车轮

铝材是易于买到，也易于加工的，选用时要考虑车轮的强度与比重(重量)的平衡。

如图 3 所示，为了充分发挥电动机带动轮胎的功能，必须尽量减轻车轮的重量(掏空)。因此应当用图 3 所示的方法。

① 平衡地打几个孔的方法。

② 不影响车轮强度的掏空方法。

在钻床上打孔的时候，若注意图 4 所示要点，就能保证不妨碍实用所需的加工精度。

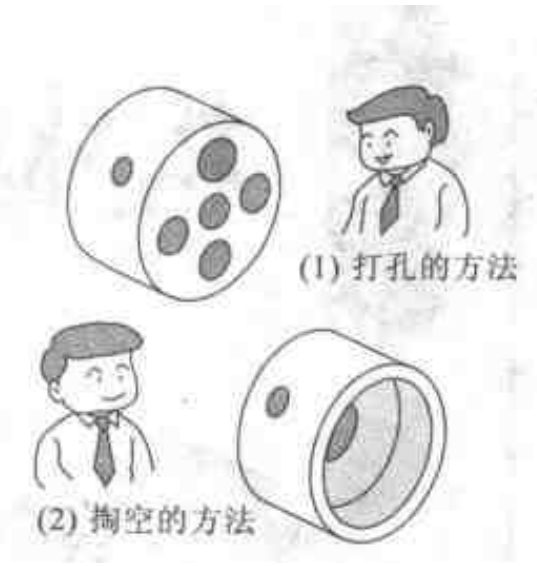


图 3　车轮的掏空(减小重量)



图 4　用钻床进行车轮的掏空加工

4 相扑机器人的组装(2)

万向轮与轮胎的种类



1 球形万向轮

前轮的用法

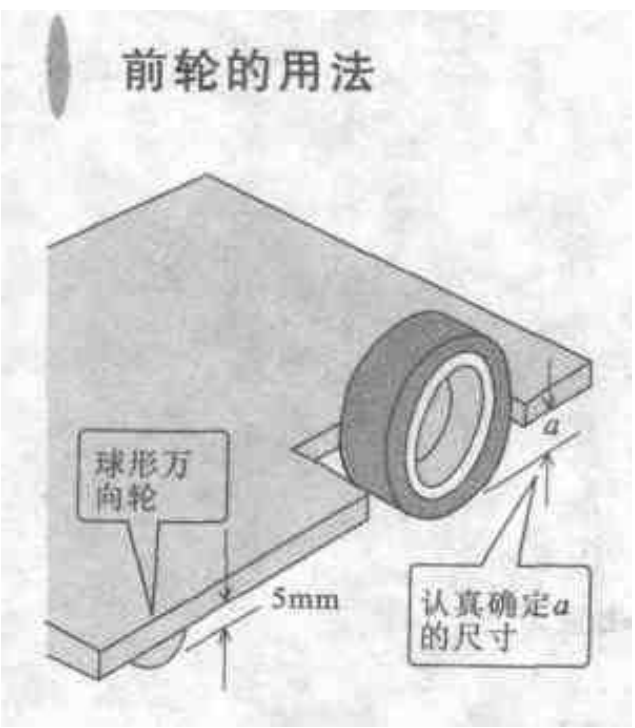


图 1　作前轮用的球形万向轮

两轴驱动的相扑机器人，其前面的车轮用 12mm 的球形万向轮。

安装在车架底面上的球形万向轮与赛场的间隙约为 5mm。因此，如图 1 所示，后部车架底面与赛场的间隙应当以 a 为基准。在决定万向轮的位置时，应着重考虑不让对方相扑机器人从前方或侧面进入。

作为“救命菩萨”的球形万向轮

如图 2 所示，当球形万向轮一滚出赛场之外，就不能用自己的力量退回，这时，图 2 所示的球形万向轮就成了“救命菩萨”。

这就是冲出去时的对策。其目的就是用一个球形万向轮也能操纵。

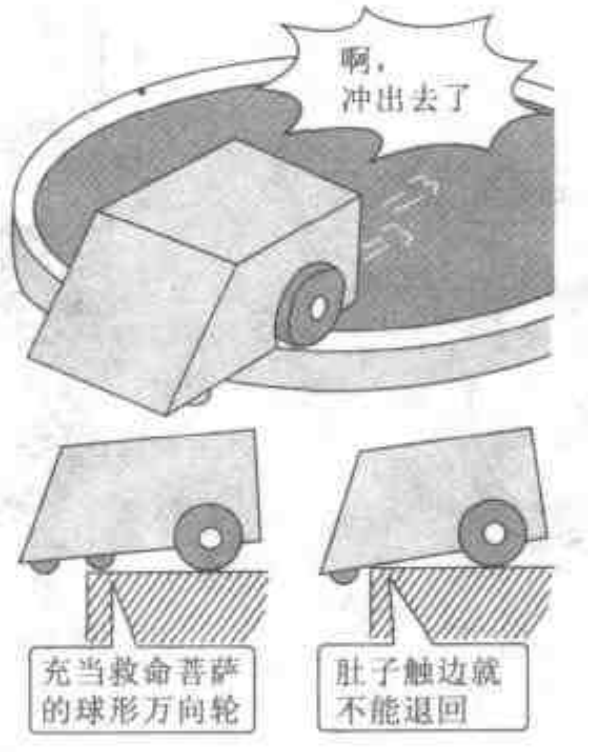


图 2　充当"救命菩萨"用的球形万向轮

本来球形万向轮是在赛场表面作为滚动用的，实际上，贴在球上的玻璃胶带是可以在赛场的橡胶表面快速地滑动的。

在使用球形万向轮方面，应当试一试。

Ⅱ　轮　胎

轮胎是将电动机的力量有效地作用在赛场上的重要部件。不管是多大的电动机驱动力加在赛场表面上，若轮胎打滑，也是没有意义的。

轮胎与车轮一样，大多使用自制的或遥控模型用的轮胎，或者试用其他工业产品的合适的轮胎状的零件。取得的途径是互相保密的。

遥控模型轮胎

遥控模型轮胎的种类，有所谓 F1 汽车模型轮胎、越野轮胎等数种。要根据轮胎的直径来选用。选择的重点应当是，为了得到尽可能大的对赛场表面的附着力。

图 3 表示遥控型轮胎的例子。

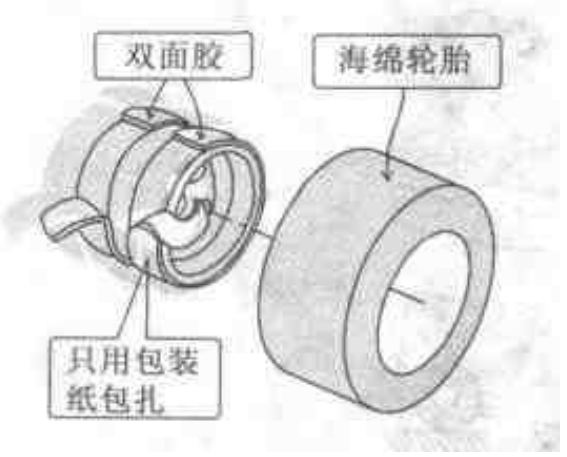


图 3　遥控模型轮胎的例子

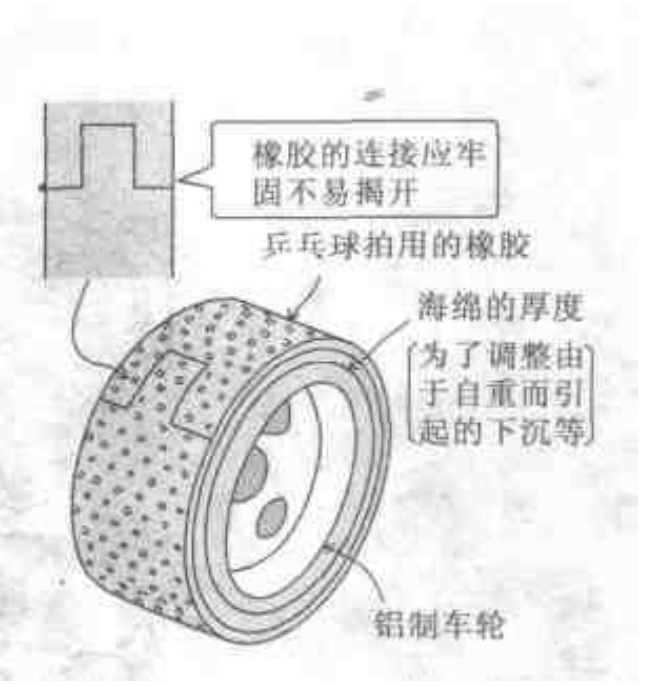


图 4　用乒乓球拍橡胶的轮胎

自制轮胎

在自己确定轮胎直径的场合，就要自制轮胎，其方法如图 4 所示，将乒乓球拍的橡胶，用双面胶等粘接材料，粘贴在自制的轮胎上。

所用的橡胶大体可分为：

- 正胶型(表面软橡胶)。
- 反胶型(里面软橡胶)。

另外，橡胶与轮胎之间，要粘贴适当厚度的海绵，也是应注意的要点。

Ⅲ　增强轮胎附着力的方法

实际在机器人相扑中，为了增大轮胎的附着力，常使用以下方法，然面无论用什么方法，都会因轮胎表面的脏物，如泥土、灰尘等，而减小附着力。



图 5　用橡胶清洗剂清洗轮胎表面

- 用乒乓球拍用的泡沫橡胶清洁剂等，涂在轮胎表面(参见图 5)。
- 把浆糊涂在轮胎表面的方法。
- 在轮胎上包裹生橡胶管的方法。

例如，把取自遥控模型轮胎上的生橡胶管，切成适当宽度，包在轮胎上。

这种场合，由于管子的结合力随着时间的推移，轮胎的直径会变小，因此必须把握好包裹管子的时间。

5 相扑机器人的组装(3)

减速机构与驱动力矩的关系



Ⅰ 减速机构

相扑机器人在赛场上的速度，取决于电动机的转速。

改变电动机的转速，用电气方法是改变电压，使用脉宽调制(PWM)控制方法。

为了有效地产生电动机的力矩，常用以下几种机械减速机构。

使用齿轮减速电动机

带齿轮的电动机，也叫齿轮减速电动机(参见图 1)。

制造厂提供各种不同减速比的电动机，可从产品样本中选用，较容易买到。

可将轮胎直接安装在齿轮减速电动机的已减速到所需转速的轴上，而不必自己制作减速机。

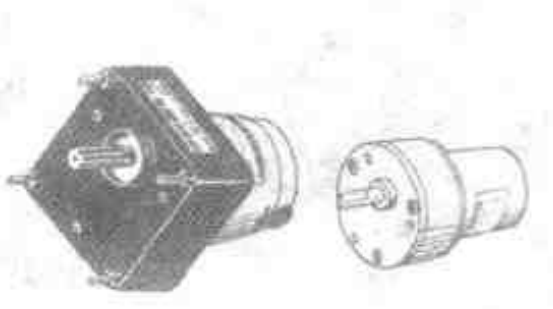


图 1　齿轮减速电动机

对初次制作相扑机器人的人来说，电动机与减速机的购置一次搞定，制作就方便了。

用伞齿轮

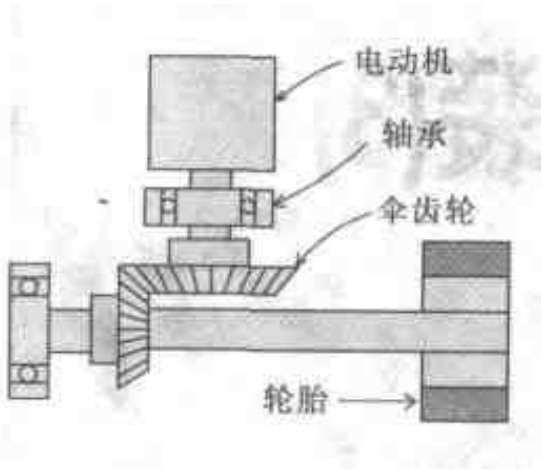


图 2　用伞齿轮减速机构

伞齿轮可使电动机轴与轮胎轴如图 2 所示，成直角位置来布置。因此可选用尺寸较大的电动机。

螺旋伞齿轮同时啮合的齿数多，可平滑地传递转矩。此时，由于螺旋会引起的垂直方向作用的推力，必须在选择轴承时予以考虑。

使用蜗轮蜗轩

也有用图 3 所示的蜗轮蜗杆减速机构的。这种方法的减速比大，但力矩的传动效率很低。另外，由于蜗轮与蜗杆的正交，装配和加工都困难。

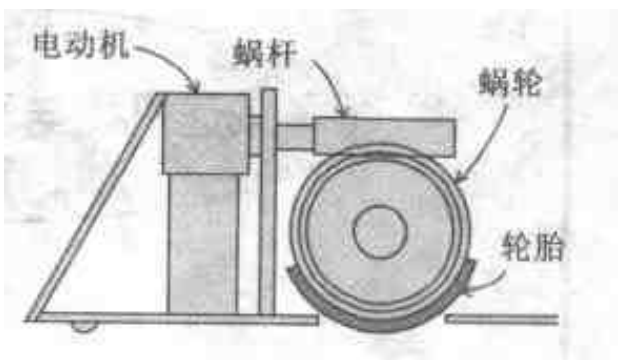


图 3　用蜗轮蜗杆的减速机构

II　驱动力与力矩的关系

相扑机器人的前进、后退、回转等操作，要靠轮胎与赛场的附着力（摩擦力），这里只考虑相对于赛场的轮胎上产生的驱动力。

在图 4 中，当轮胎被来自电动机的转矩 T 驱动时，在赛场与轮胎接触点上产生力 F。

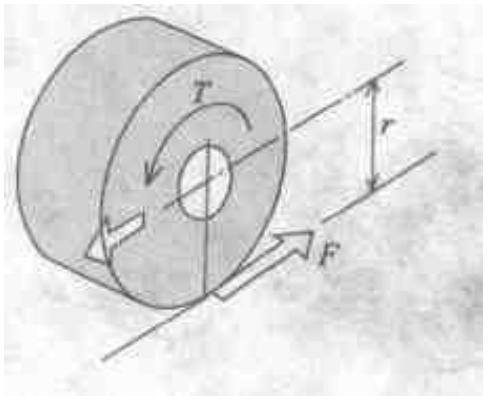


图 4　轮胎产生的驱动力

这个力是轮胎对赛场的挤压力（机器人的操纵力），其大小与力矩成正比，与车轮半径成反比。可用下列公式表示：

$$F=\frac{T}{r}$$

式中　F—— 驱动力(N)；

T——转矩(N·m)；

r—— 轮胎半径(m)。

由此公式可知，要得到大的驱动力，就必须有大的转矩。

下面来讨论加大转矩的方法。

增大转矩的方法

再用上述公式，来考虑一下产品样本中转矩的定义，简单说来，“转矩 T 可看作是旋转力矩的大小”。

现在，在图 5(a)所示的有 60×10^{-4}N·m 转矩的直流电动机(额定转速为 8000r/min)上，安装 6cm 的轮胎时，试求其驱动力。

则力为

$$F=T/r=(60\times10^{-4})/0.03$$
$$=0.2(\mathrm{N})$$

然后再考虑一下图 5(b)所示的，在电动机轴上安装 1cm 的齿轮，用 6cm 齿轮减速后的轮胎所产生的力 F，则力为

$$F=60\times10^{-4}/0.005=1.2(\mathrm{N})$$

这个力的大小，是通过齿轮传递的。

现在若求 6cm 的齿轮的转矩，则为

$$T=1.2\times0.03=0.036(\mathrm{N}\cdot\mathrm{m})$$

是额定转矩的 6 倍。

再求此时轮胎的驱动力为

$F=0.036/0.03=1.2(\mathrm{N})$。

也是额定驱动力的 6 倍。

但此时的转速则是额定转速的 1/6，成了 1 333r/min。

这一转速之比，称为减速比。

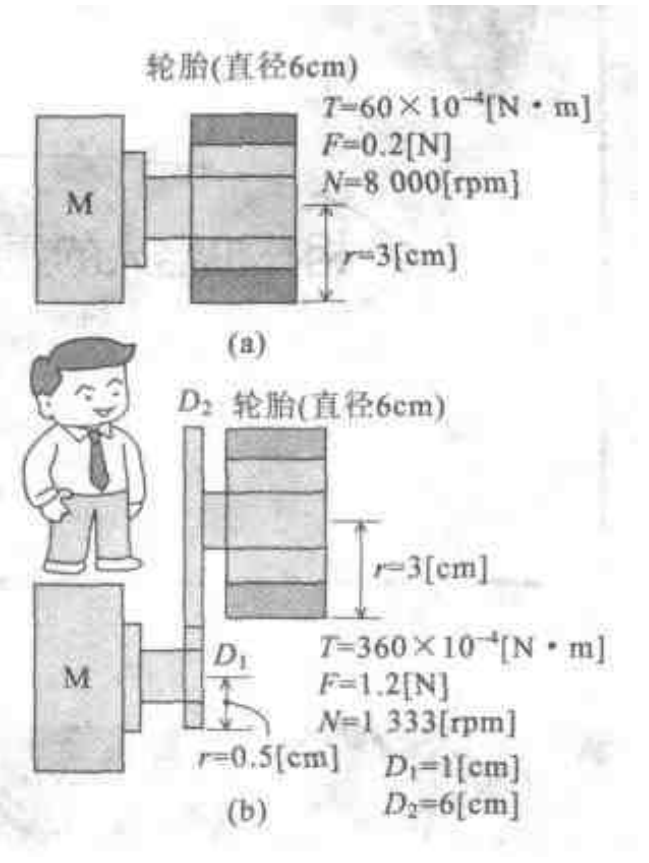


图 5　驱动力与转速的关系

现在，若使用 1/30 的齿轮减速电动机，则转速成为 1/30，驱动力则成为 30 倍。因此，力量大的相扑机器人，是通过减速机构对电动机进行减速，才得到大的驱动转矩的。汽车在启动时，使用低速齿轮，也是相同的原理。

当然，无论用多大的减速比，电动机本身的输出功率并没有增加。

6 相扑机器人的组装(4)

相扑机器人的重心与驱动速度



I 相扑机器人的重心位置

因为要靠有效利用相扑机器人的重量来比赛，所以相扑机器人的重心位置是很重要的。

前后的重心位置

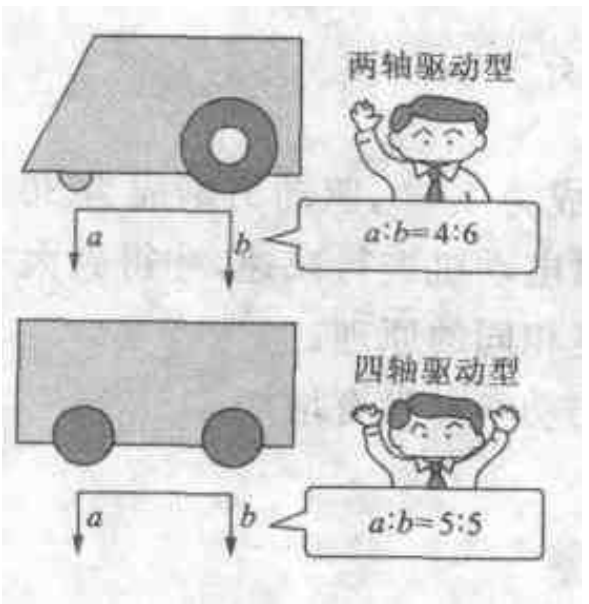


图 1　相扑机器人的前后重心位置

两轴驱动的两个球形万向轮与两个电动机的场合，往往认为重心越靠近电动机端，则所产生的动力越大。

但是，这种场合，重心当然是移往后方的。因此，当对方相扑机器人从正面攻击时，容易造成抬起的“担心”状态，就是被对方钻入而被打败的结局。

同时还必须防止对手从横向与侧后方的攻击。根据一些有制作相扑机器人经验的人士的说法，应如图 1 所示，“以前轮 4、后轮 6 的比例”来分配重量。

重心高度的位置，与“相扑运动员的秘诀：把腰压低”一样，希望降低重心。

左右方向，则重心应当尽量取中间部位，加大球形万向轮轮胎间隔，以提高稳定性，也是要紧的。

重心位置的简易测量法

所制作的相扑机器人的大概重心位置，可用图 2 所示的方法来求得。

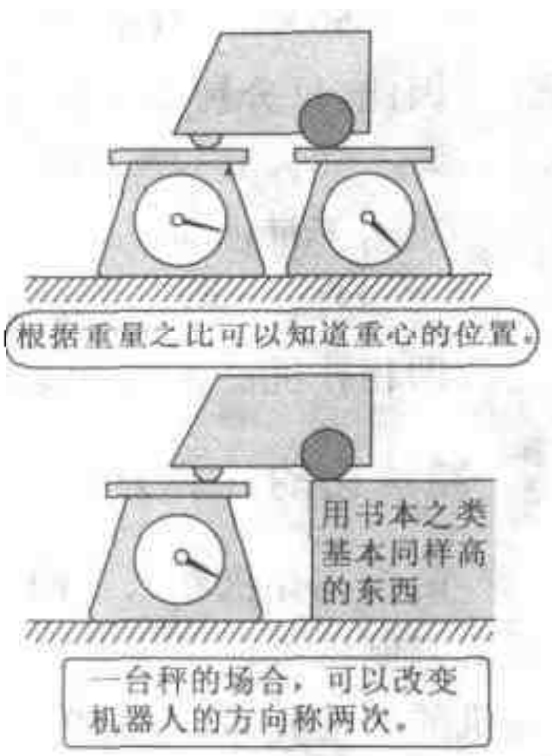


图 2　重心位置的测量方法

重心位置的调整

重心位置的调整，一般可用电动机、电池块等能比较简单地改变位置的零件。

另外，可用片状小铁块等，铁块的大小可以从铁的比重为 7.8kg/cm² 计算出来。

Ⅱ　相扑机器人的驱动速度

相扑机器人的驱动速度的调整依据为：要以“干吧！加油”分隔线中冲出去，在 1.54m 赛场中自由地来回运动，并且不得冲出白色的赛场边界线，获得与对方相扑机器人进行战斗所必须的动力。

相扑机器人的速度计算方法

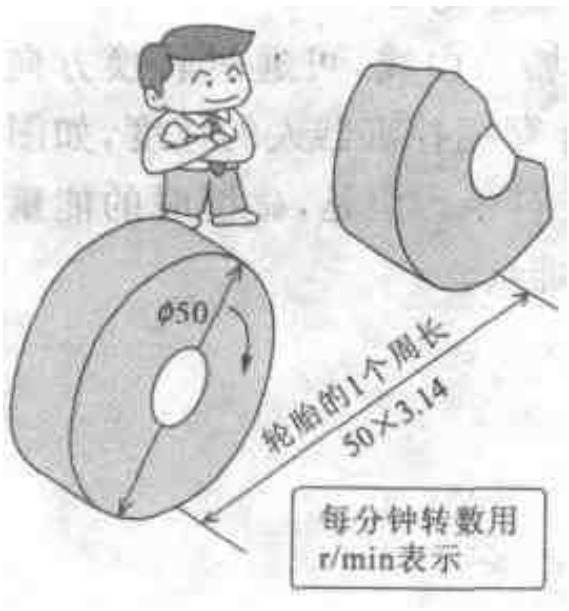


图 3　轮胎每转一圈的移动距离

从

距离＝速度×时间

的关系中可知

速度＝距离÷时间

1 秒的速度＝轮胎外围圆周长度×每分钟转速/60

例题

试求轮胎的直径为 50mm，200r/min（每分钟的转速）相扑机器人的速度。

轮胎转一圈或一个圆周所移动的距离，其长度如图 3 所示为

50×3.14(mm)

因此，每分钟 200 转，则为 200 倍。

50×3.14×200(mm)－31 400(mm)

相当于每秒为

31400÷60＝523(mm)

即相扑机器人的速度为每秒 0.523m。

减速比的计算方法

制作相扑机器人在确定轮胎尺寸时，主要考虑要受到 200cm×200cm 底面积的限制就可以了。此时，通过自己所设计的相扑机器人的速度，就可以从电动机的额定转速，求出必要的减速比。

「例」

轮胎	直径 40mm
相扑机器人速度	80cm/s
电动机的额定转速	4 500r/min

80 cm/s 速度为每分钟

80×60 ＝4 800(cm/min)

＝48 000(mm/min)

以此速度求得轮胎直径为 40mm 的每分钟速度为

N ＝ 48 000/(3.14×40) ＝ 381/(r/min)

必要的减速比为

i ＝381/4 500＝1/11.8

这种场合就取减速比 1∶12。



图 4　相扑机器人的最高速度

传感器检测赛场的白线，可通过转换方向用的反应时间，来计算相扑机器人的速度，如图 4 所示，每秒 1m 称为临界速度，若速度过快，则可能自杀。但是，碰撞时的能量与相扑机器人的速度有关，所以要二者兼顾就是其难点。

7 相扑机器人的控制

控制系统与CPU



I 控制系统

可以认为:“智能型相扑机器人的控制方法没有限制”。与无线遥控型不同,因为智能型相扑机器人的控制操作,要靠自己判断,所以要把控制系统装在机器人自身的机座上。

也就是说,控制系统需要有相当于人们所说的判断用的头脑,相当于取得信息用的五官,CPU、传感器以及使它们起作用的能源。

智能型相扑机器人有着无线遥控型所没有的趣味。

智能型相扑机器人的控制系统概要如图 1 所示。

II 单片机

控制计算机常用单片式微型计算机(简称单片机)。单片机的定义如图 2 所示,以 8 位作为代表的 CPU 有 Z80 等,是将微型计算机所要用到的全部功能,以及作为外围集成电路(IC)的并行输入输出接口(PIO)、时钟控制器(CTC)、串行输入输出口(SIO)等,均集成到一片集成电路(IC)中,即称为单片机。也有 16 位的单片式微型计算机。

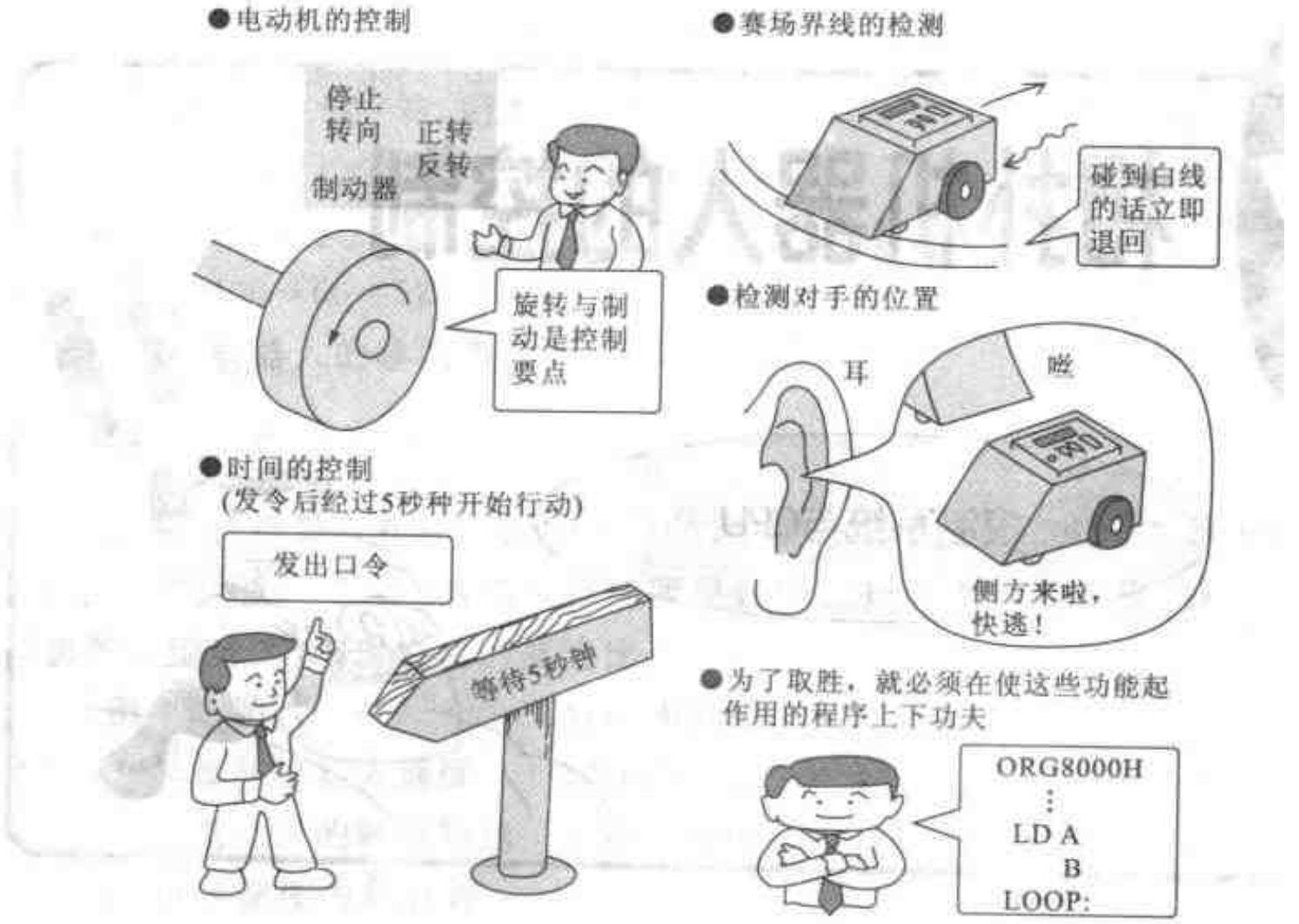


图 1　智能型相扑机器人的控制系统

由于单片机体积小，已广泛用于其他的电子设备或产业用的控制设备中。

在单片机中含有只读存储器(ROM)或读写存储器(RAM)等功能的一体化卡片式计算机，如图 3 所示，常用于相扑机器人的控制中。

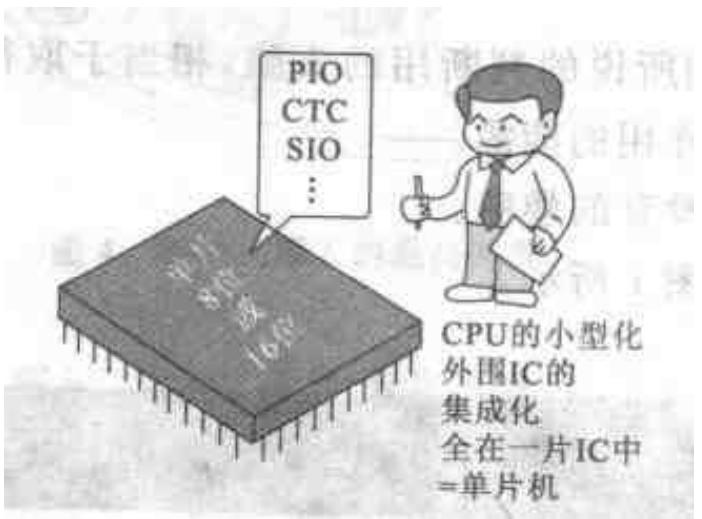


图 2　单片式微型计算机

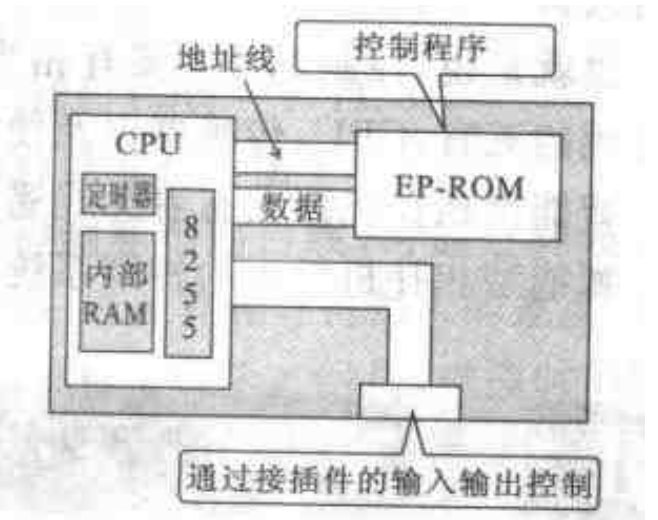


图 3　卡片式计算机的结构

Ⅲ 单片机的编程

单片机的源程序，是用图 4 所示的个人计算机的编辑器编制的。

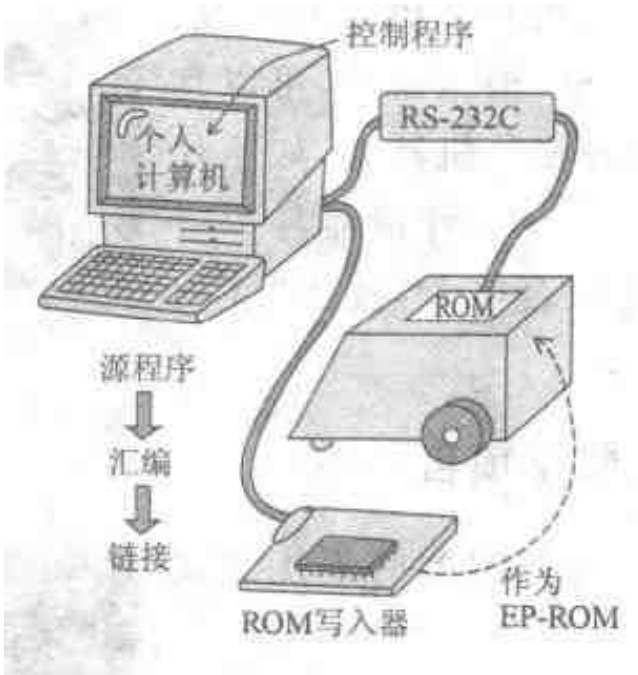


图 4　单片式微型计算机的编程

若将单片机与 RS-232C 电缆连接，可通过所编制的程序，在个人计算机的监视器上进行控制。因此，必须有编辑、汇编、链接等开发用的支持软件系统。编制好的程序也可简单地转录、修改，验证程序的工作也很容易。

通过这些程序，只要按下按钮开关等，起动 ROM 中的程序，用来自光传感器等的信号，即可进行电动机启动等控制。

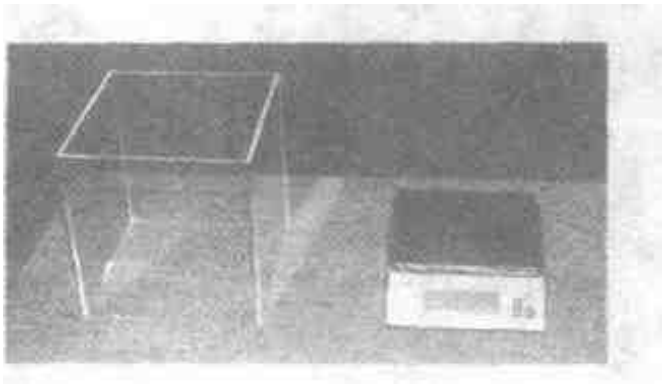
一般来说，要通过 ROM 写入器将程序写入 EP-ROM 中，采用装入卡片式计算机的控制方法。

无论何种控制方法，都是通过数个能用于控制的输入输出口，以信号交换的方式，来对执行机构的电动机等进行控制的。

相扑机器人比赛大会规则摘要

比赛的方法

三场两胜制，先胜两场的相扑机器人取胜。比赛时间 3 分钟。有加时赛，根据需要，可以判断、抽签决定胜负。得胜的相扑机器人，如照片 1 所示，要用 200mm×200mm 的计量器与数字秤校核其大小与重量，验证合格者，授予“胜利名誉”(参见照片 1)。



照片 1　相扑机器人的计量器与秤

相扑机器人的主要禁止项目

① 禁止设置干扰对方控制器的干扰波发生装置。

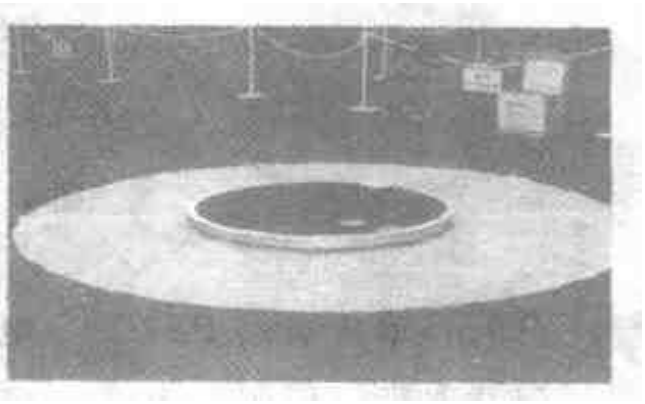
② 禁止使用损坏赛场的部件。

③ 不得内置液体、气体、粉末往对方相扑机器人喷射。

④ 不允许有扔、洒物品的装置。

⑤ 不得用吸盘、浆糊等将相扑机器人固定在赛场上。

更详细的资料，可查阅机器人相扑大会的规则，赛场大小等的规定。



照片 2　关东地区比赛大会的赛场与外围空地。

赛场大小等的规定

图 1 表示赛场与铺砂的主要尺寸。

直径：1.54m

高度：5cm

材质：黑色硬橡胶

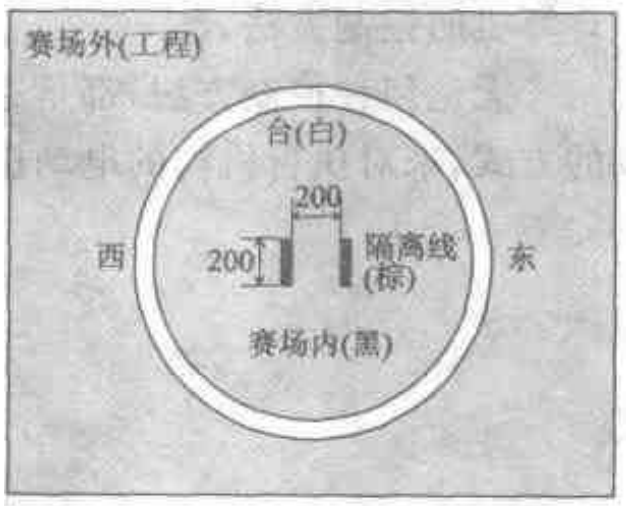
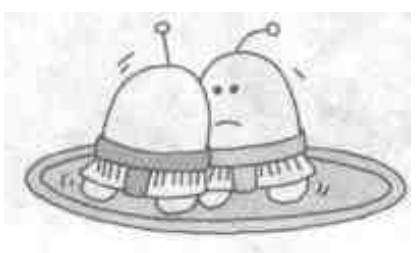


图 1



赛场的台(圆环)：宽 2cm 长 20cm，棕色线。

★赛场外围(空地)，也有圆形的，见照片 2。

译者版

进入21世纪，尤其我国加入WTO之后，科学技术、工农业生产迎来了新的发展机遇，也面临着新的挑战。

为了适应在国际大市场上竞争的需要，各行各业的人们都迫切需要学习新技术、新理论、新方法。机电一体化正是所有人都多少会接触到的实用技术。它广泛深入到从军用到民用、从工业到农业的所有国民经济领域，小到手表、照相机，大到汽车、飞机、火箭、人造卫星、核电站等，无不应用了机电一体化的技术和产品。

《机电一体化》这本书，是满足各行各业人们需要的一本入门教材。书中对原理的讲述深入而详细，图文并茂，文字浅显易懂，并有练习题、挑战题，便于复习掌握要点。非常适合作为中高级技师、就业培训及军地两用人才的培训教材，也可以供有初中以上文化程度的有兴趣的读者自学之用。也是理工科中等专业学校的教师们了解这方面知识的必备教材或参考手册。本书的编写既全面地描述了机电一体化的各方面技术，充分阐述其基本原理，又有非常实用的例子，适合实际制作，并且各章节中一个一个专题均自成体系，相对独立。适合于不同层次的读者任意选学。由于本书的上述特色，其原版书十分畅销并已多次重印，深受读者欢迎。

本书中所提及的内容，既有当前正在普及的技术，也有一些比较新颖的内容，如最后部分介绍的机器人的制作等。读者可在掌握本书内容的基础上，进一步学习自己感兴趣的相关新技术，努力跟上时代的前进步伐。